AD-A038 135 GTE SYLVANIA INC MOUNTAIN VIEW CALIF ELECTRONIC SYST--ETC F/G 17/9
ADVANCED ARRAY DESIGN, TEST AND EVALUATION.(U)
JAN 77 J E RICE

UNCLASSIFIED

F30602-74-C-0252

RADC-TR-77-25

NL

1 OF 2
AD A038135

RADC-TR-77-25
Final Technical Report
January 1977

ADVANCED ARRAY DESIGN, TEST AND EVALUATION

GTE Sylvania Incorporated

ROME AIR DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
GRIFFISS AIR FORCE BASE, NEW YORK 13441

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

This report has been reviewed and approved for publication.

APPROVED: Robert A. Mather

ROBERT A. MATHER
Project Engineer

APPROVED: Joseph L. Ryerson

JOSEPH L. RYERSON
Technical Director
Surveillance Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: RADC-TR-77-25	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): ADVANCED ARRAY DESIGN, TEST AND EVALUATION	5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report. 1 May 74 – 30 June 76
6. PERFORMING ORG. REPORT NUMBER: N/A	
7. AUTHOR(s): James E. Rice, Jr.	8. CONTRACT OR GRANT NUMBER(s): F30602-74-C-0252, and now F30602-75-C-0287
9. PERFORMING ORGANIZATION NAME AND ADDRESS: GTE Sylvania Incorporated, Electronic Systems Group, Western Division, Mountain View CA 94042	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62702F 12417F 65121606 414L0125
11. CONTROLLING OFFICE NAME AND ADDRESS: Rome Air Development Center (OCSL), Griffiss AFB NY 13441	12. REPORT DATE: Jan 77
	13. NUMBER OF PAGES: 122
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Same	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

DDC RELEASED APR 7 1977 C

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Robert A. Mather (OCSL)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Antenna Design Array
Antenna Test and Evaluation
Antenna
High Frequency Planar Array
Computer Controlled Antenna

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

From 1 May 1974 to 30 June 1976, GTE Sylvania, Incorporated, Electronic Systems Group-Western Division, provided engineering field services in support of the RADC experimental HF FM/CW backscatter system at Dexter, New York. These services were primarily concerned with the design and development of a low side-lobe planar array, computer control of system parameters, and prediction of performance from precision measurements. This report discusses and details these efforts.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

406 571

PREFACE

The work described in this Final Report was performed by GTE Sylvania, Incorporated, Electronics Systems Group, Western Division, P.O. Box 205, Mountain View, California. It was accomplished under the direction of Rome Air Development Center, Griffiss Air Force Base, New York, under Contracts F30602-74-C-0252 (Project 6512) and F30602-75-C-0287 (Project 414L).

Numbered by the contractor, M1601, the report covers the period 1 May 1974 to 30 June 1976.

Mr. John T. Clancy and Mr. Robert A. Mather, both of OCSL, were the RADC contract monitors.

Acknowledgement is given to Mr. Clennon Moore for his invaluable suggestions during the design and construction of the system.

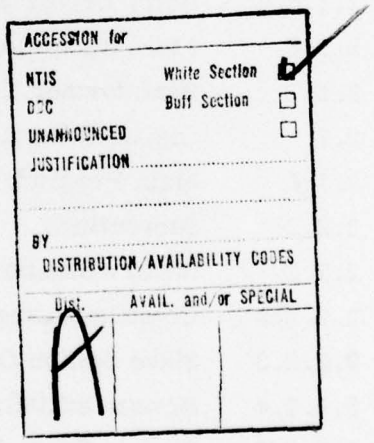


TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF TABLES

EVALUATION

This effort was part of the RADC Over-the-Horizon Technology Program which was funded by and in support of the ESD 414L Prototype Radar System (PRS). It represents a major part of the low sidelobe planar array development which was initiated under this program to provide the PRS with a technology base from which to derive final design parameters.

Specifically, this effort was concerned with the design of the computer control logic and software design, implementation on a PDP-11/40 computer system, and development of a computer model of the planar array that would incorporate measured antenna patterns and error functions. The ultimate goal was to determine the feasibility of a computer controlled extremely low sidelobe array, and the prediction of system performance and degradation as the sidelobe levels degraded.

Unfortunately the program was terminated before meaningful data could be collected and the system could be evaluated.

The computer controlled functions and preliminary breadboards, as far as investigated, operated as expected. A reasonable conclusion is that the array, if completed and properly tested, would have met original design goals.

ROBERT A. MATHER
Project Engineer

Section 1

INTRODUCTION

Since 1969, GTE Sylvania has provided field engineering services at RADC concerned with the investigation, development, and implementation of advanced data processing, analysis techniques, and HF array design, directed toward the test and evaluation of the RADC experimental HF FM/CW backscatter system in upstate New York.* The research and development activities during the period 1 May 1974 to 30 June 1976 included the following.

a. Design of computer-controlled relay switching, synthesize control and function storage units.

b. Computer-controlled communication system between a master station and remote slave units.

c. Software for control and testing of the digital control systems.

d. Design of a precision calibration unit and technique.

e. Array modeling to use measured and random errors to predict array performance.

f. Limited beam synthesis to obtain aperture coefficients for arbitrary beamwidths and shapes.

These activities were in support of an effort to construct a large planar array at the RADC Dexter, N.Y., site. This array was to have been an experimental computer-controlled array with a design goal to provide 80 dB sidelobes with both azimuth and elevation steering.

---

*A working knowledge and understanding of the RADC FM/CW system is assumed. Also, it should be noted that some of the results presented in this report are preliminary or transitory, since the system has not attained final operational status.

Engineering support was provided for overall system design, breadboard and component selection testing, and flight testing. However, the primary engineering support of the system was the logical-and-hardware design of those subsystems enclosed within the dashed line shown in Figure 1. In addition, the engineering support effort was to provide all software systems for the PDP 11/40 computer, including both the logic control and signal processing.

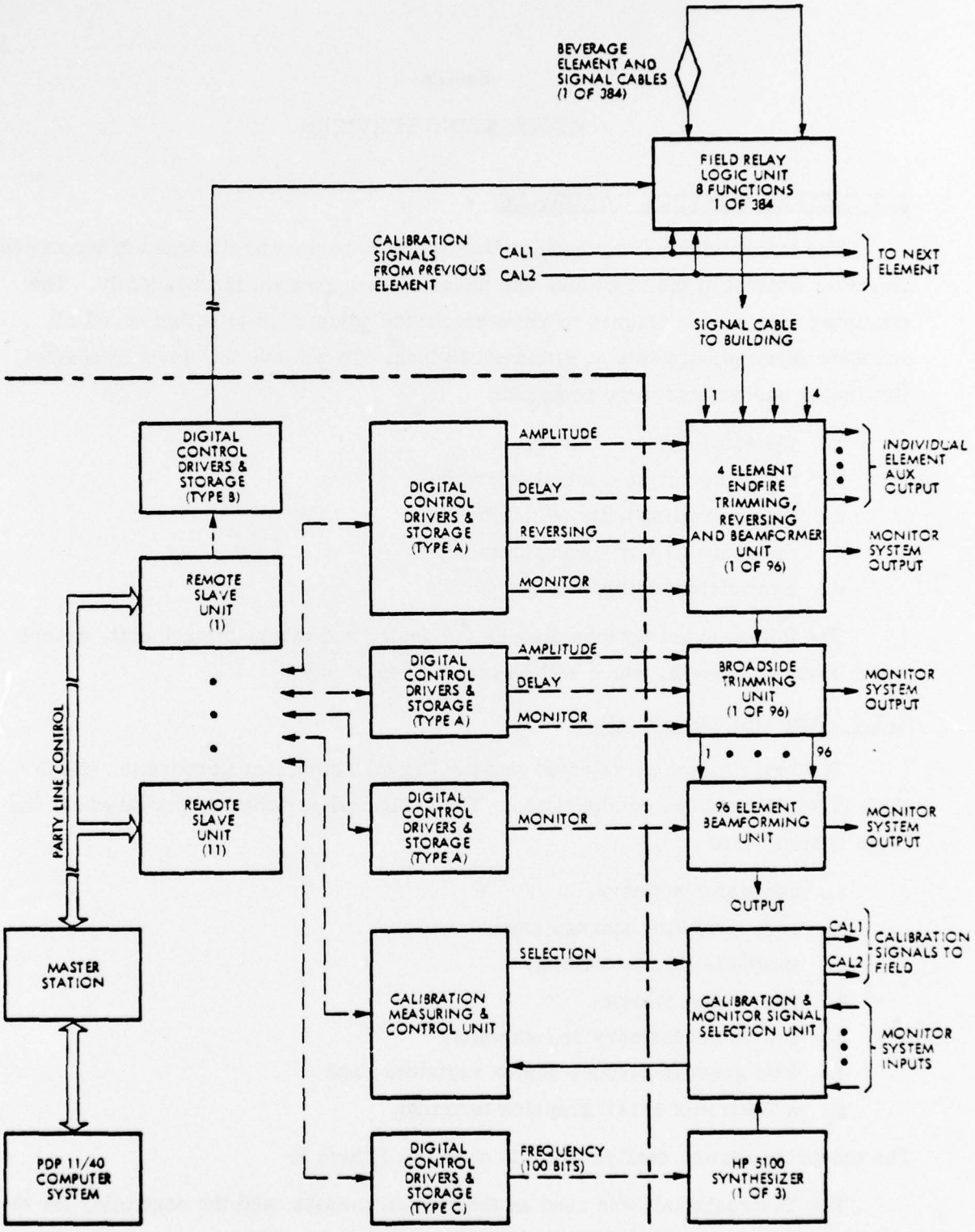


Figure 1. Computer Controlled Planar Array Block Diagram

Section 2

ENGINEERING SERVICES

2.1 DIGITAL CONTROL HARDWARE

The low sidelobe array built at Dexter, New York, was designed to incorporate computer control of the amplitude and phase of each element independently. The trimming units were designed to allow amplitude adjustment to within ±0.03 dB and time delay adjustments to within ±0.05 inch. To achieve this level of control, five major subsystems were required:

a. the computer,
b. the computer-to-control interface,
c. the controlling units (with storage),
d. the controlled trimming units, and
e. a precision calibration system.

The following paragraphs discuss the design and theory of each unit, except the calibration system, which will be treated separately.

2.1.1 Computer Configuration

The basic computer selected was the Digital Equipment Corporation (DEC) PDP 11/40 in the GT44 configuration. The additional peripherals procured for the basic system were

a. additional memory,
b. additional disk storage units,
c. magnetic tape recording,
d. paper tape system,
e. two direct memory access units,
f. four general-purpose duplex registers, and
g. a Tektronix 4014E graphics terminal.

The computer system configuration is shown in Figure 2.

The Tektronix unit was used as the system console, and the controller for the LA-36 DEC writer was modified (by DEC) to allow the LA-36 to substitute as a line printer.

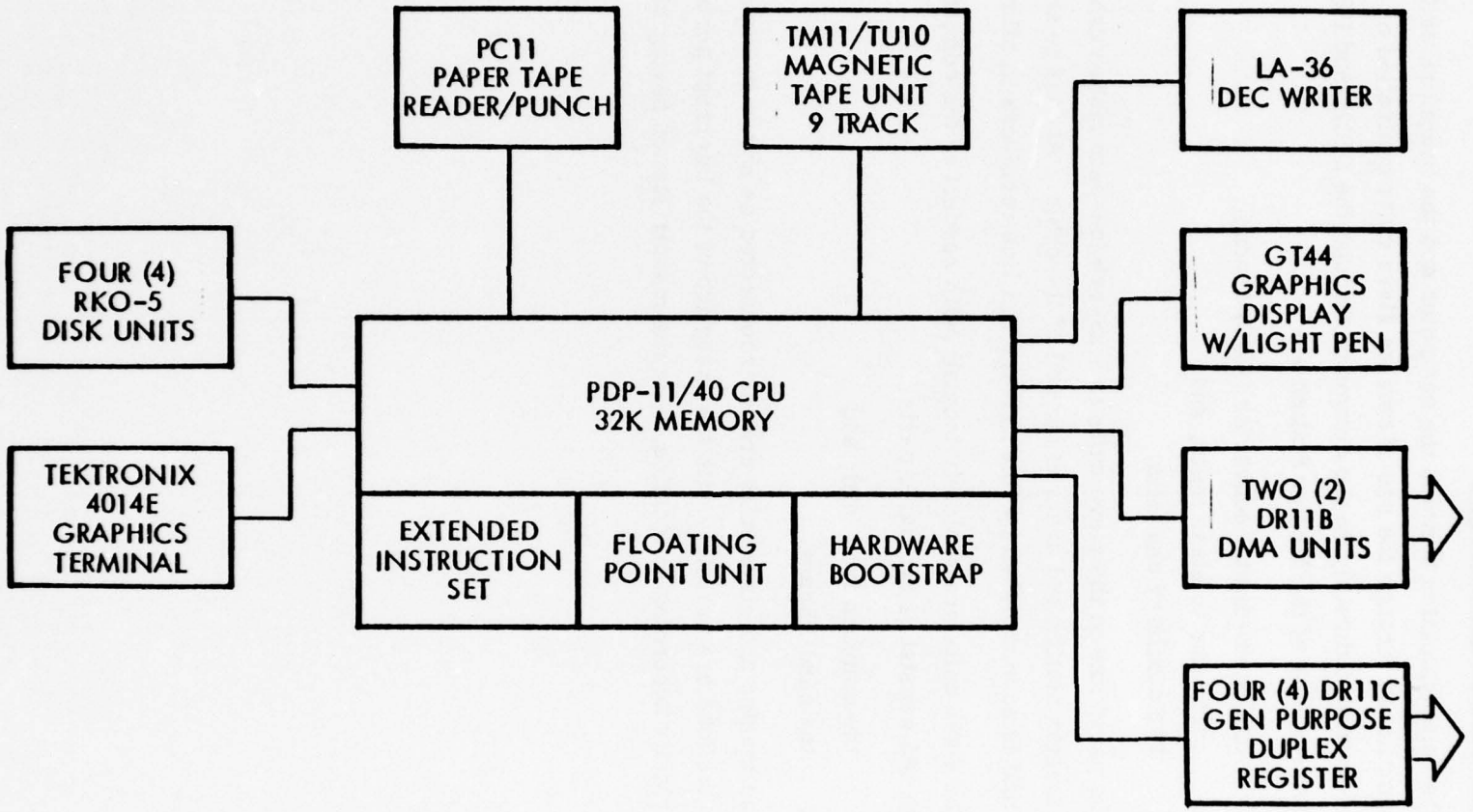


Figure 2. PDP 11/40 Computer Configuration

2.1.2 Communications System

The communication between the computer and the beamformer trimming units is accomplished through the use of remote slave units controlled by a master station connected directly to the computer through the DR11B and DR11C registers. The input/outputs of the Master Station are

a. analog-to-digital converter (ADC) channel,
b. party line to slave units, and
c. two auxiliary channels.

The party line to the slave units is a 28-twisted-pair ribbon cable terminated at the master station and again at the end of the cable. Access to and from the party line is accomplished by line drivers and line receivers in all cases.

The slave units are remotely located (with respect to the computer master station) and consist of two basic parts:

a. the common front end, and
b. the card library.

The master and slave units are interconnected as shown in Figure 3. Note all connections are parallel, with the exception of the interrupt permit flag, which sets priority according to location, with the master station having highest priority.

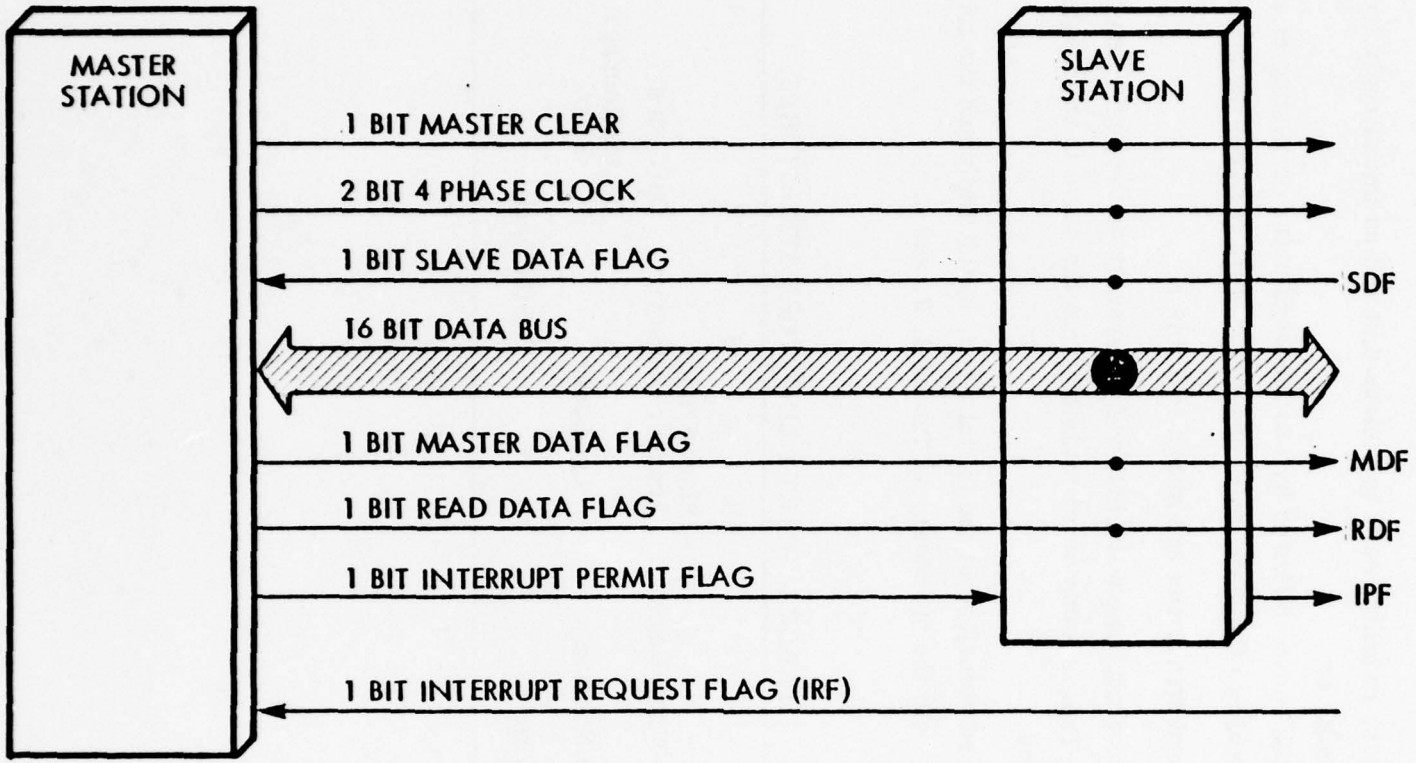


Figure 3. Party Line Configuration

2.1.3 Party Line Word Organization

The communication between master and slave units consists of either

a. just an initial word to a slave unit and an initial reply from the slave unit, or

b. the above, followed by 1 to 100 word pairs consisting of a device number word to the slave unit, and a data word to/from the slave unit.

The word structures are shown in Table 1.

The one remaining word that will be present on the party line is the slave unit interrupt. This is a one-word transfer from the slave to the master then to the computer.

The word structure of the initial word, initial reply and the interrupt word are shown, with the op-codes, in Tables 2, 3, and 4.

TABLE 1. PARTY LINE WORD STRUCTURES.

Bit	15 8	7 0
Initial/Word/Reply	Slave Station Address (Binary)	Op-Codes
Device Number	Op-Codes	Device Number (BCD)
Data Word	16 Bits Data	

TABLE 2. INITIAL WORD INPUT TO ALL SLAVE STATIONS

Bit		Description
0		Local reset required (1)
1		(1) = communications check
2		Data transfer (1)
3		Input to slave (1), output from slave (0)
4		Interrupt (1), no interrupt (0) at completion
5		
6		
7		
8	2^0	Address of slave station requested
9	2^1	
10	2^2	
11	2^3	
12	2^4	
13	2^5	
14	2^6	
15	2^7	

TABLE 3. INITIAL REPLY BY ALL SLAVE STATIONS

Bit		Description
0		Busy
1		
2		
3		
4		
5		Error; cannot execute instruction
6		Local error detected
7		Acknowledge instruction
8	2^0	Address of slave station answering
9	2^1	
10	2^2	
11	2^3	
12	2^4	
13	2^5	
14	2^6	
15	2^7	

TABLE 4. INTERRUPT WORD BY ALL SLAVE STATIONS

Bit		Description
0		Reserved
1		
2		
3		
4		
5		Error; could not complete instruction
6		Local error detected
7		Completed last instruction
8	2^0	Address of slave station interrupting
9	2^1	
10	2^2	
11	2^3	
12	2^4	
13	2^5	
14	2^6	
15	2^7	

2.1.4 Master Station

The master station (see Figure 4) is composed of:

a. six DR11 interface units,

b. seven master station interface units, and

c. three data/logic bus lines.

Data Bus 1 is composed of the individual outputs (control and data) of all six DR11 units, which are then made available as parallel inputs to all master station interfaces.

Data Bus 2 is composed of the individual outputs (control and data) of all seven master station units, which are then made available as parallel inputs to all DR11 interfaces.

The source selection bus is composed of parallel logic lines that allow any DR11 to control the input source, from data bus 1 and 2, for each of the 13 interface units.

2.1.4.1 Source Selection

This configuration was chosen to allow flexibility in the selection of computer hardware, register (software transfers), or direct memory access (hardware transfers) of data to and from interfaces. Therefore, to make a complete circuit, two source selections must be made to interconnect two units. It should be noted under this system that no two master station interfaces can be interconnected, nor can any two DR11's be interconnected.

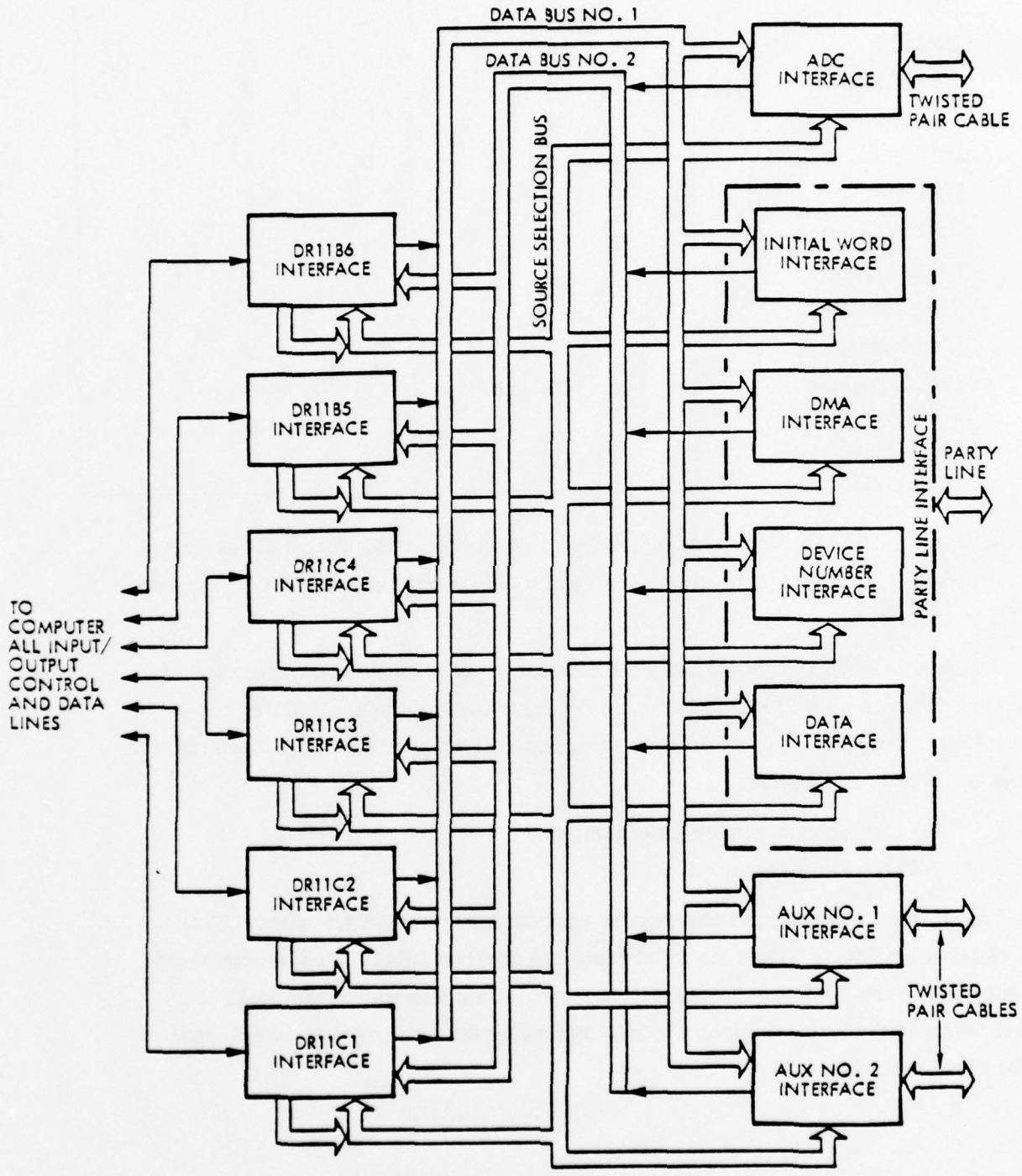


Figure 4. Master Station Interconnection, Block Diagram

TABLE 5. INTERFACE UNIT CODING

Station	Address	Select Code
DR11C1	1	1
DR11C2	2	2
DR11C3	3	3
DR11C4	4	4
DR11B5	5	5
DR11B6	6	6
A/D Converter	7	1
IW Interface	8	2
DMA Interface	9	3
DN Interface	10	4
Data Interface	11	5
AUX 1	12	6
AUX 2	13	7

The address is used to determine which DR11 or station is to utilize the source select code. The select code is used to select the source input for the addressed DR11 or station.

To connect a DR11 to a station so that the flow of information can go either way, two "source selection words" must be generated by the DR11 doing the connecting. (Any DR11 can be used to connect itself or another DR11 to a station.) These two words are called

a. master station source selection, and
b. DR11 source selection.

The first word transmitted is the master station source selection word. This will cause a station to select its input from the desired DR11. The second word transmitted is the DR11 source selection word. This will cause the DR11 to select its input from the desired station, normally the unit previously selected in the paragraph above.

The DR11 and station thus connected will remain this way until they are reconnected (as determined by the computer program) or until the power is removed.

2.1.4.2 Source Selection Word Structure

Each DR11 outputs 16 bits (2^0 - 2^{15}), exclusive of control bits, of which the master station utilizes 9 bits (2^0 - 2^8) for source selection. In addition, two control bits are used for control and timing. The structure of these words (bits 2^0 - 2^8) are as shown in Table 6.

TABLE 6. SOURCE SELECTION WORD STRUCTURE

	2^0	2^3	2^4	2^6	2^7	2^8
Station	ADDRESS OF STATION		SELECT CODE OF A DR11		ENABLE	RESET
DR11	ADDRESS OF A DR11		SELECT CODE OF A STATION		ENABLE	RESET

Note: The word (bits 2^0 - 2^8) described above must be in one's complement before transmitting to the master. Before the one's complement is performed, bit 7 is defined as $\underset{\text{Low}}{0}$ = ENABLE.

The words (bits 2^0 - 2^8) are inputs to a source selection bus control (SSBC) unit in the master station. Each DR11 has control of one SSBC unit. The parallel outputs of these SSBC units share a common bus line.

To enable an SSBC unit, the DR11 initiating the connection (only one in any given sequence) must set the first control function to a high level, after the source selection word is available. The second control function is then set and cleared before the source selection word and first control function are removed. The control functions used are shown below.

Function	1	2
DR11B	FNCT3	FNCT2
DR11C	CSR1	CSR0

2.1.4.3 Source Selection Recorder

Each DR11 and station has source selection decoder (SSD) units, and all SSD inputs are connected in parallel. The input to the SSD units is from the parallel output of the SSBC units (described).

The SSD units consist of four parts:

a. An address comparator,
b. Source selection storage,
c. Enable/disable storage, and
d. A reset gate.

The address comparator uses bits $2^0 - 2^3$ to determine if the DR11 or station is being addressed and, if true and enabled, will enable the source selection storage and reset gate. The second computer function from the SSBC units is used as the enable for the address comparator.

The source selection storage will store bits 2^4 - 2^7 only after being enabled by the comparator. Bits 2^4 - 2^6 (select code) will then cause the DR11's or the master station source multiplexing units to select the correct source, if enabled. Bit 2^7, when low, will enable the DR11 or station source multiplexing units to pass the data or, when high, will disable, causing a constant low output.

The reset gate inputs are from the address comparators and bit 2^8 and last only during the time the second computer function from the SSBC is high. Note that the reset pulse, if initiated in the initial word interface, will reset all party line interface units ; however, if the reset pulse is initiated in any other interface, it will reset only that particular interface.

2.1.4.4 Party Line Interface

2.1.4.4 Party Line Interface

There are eight functions of the party line interface unit:

a. Receive an initial word from the computer.
b. Establish communication with the requested slave.
c. Transmit the initial word to the slave units.
d. Receive the initial reply from the slave unit.
e. Transfer data to or from the slave unit.
f. Receive interrupts from the slave units.
g. Interpret any error conditions and send to the computer.
h. Perform communication checks when desired.

The party line interface, in addition to the above, will operate and appear transparent to the computer, such that the computer can be used as if connected directly to the selected slave unit.

The block diagram of the party line interface is shown in Figure 5.

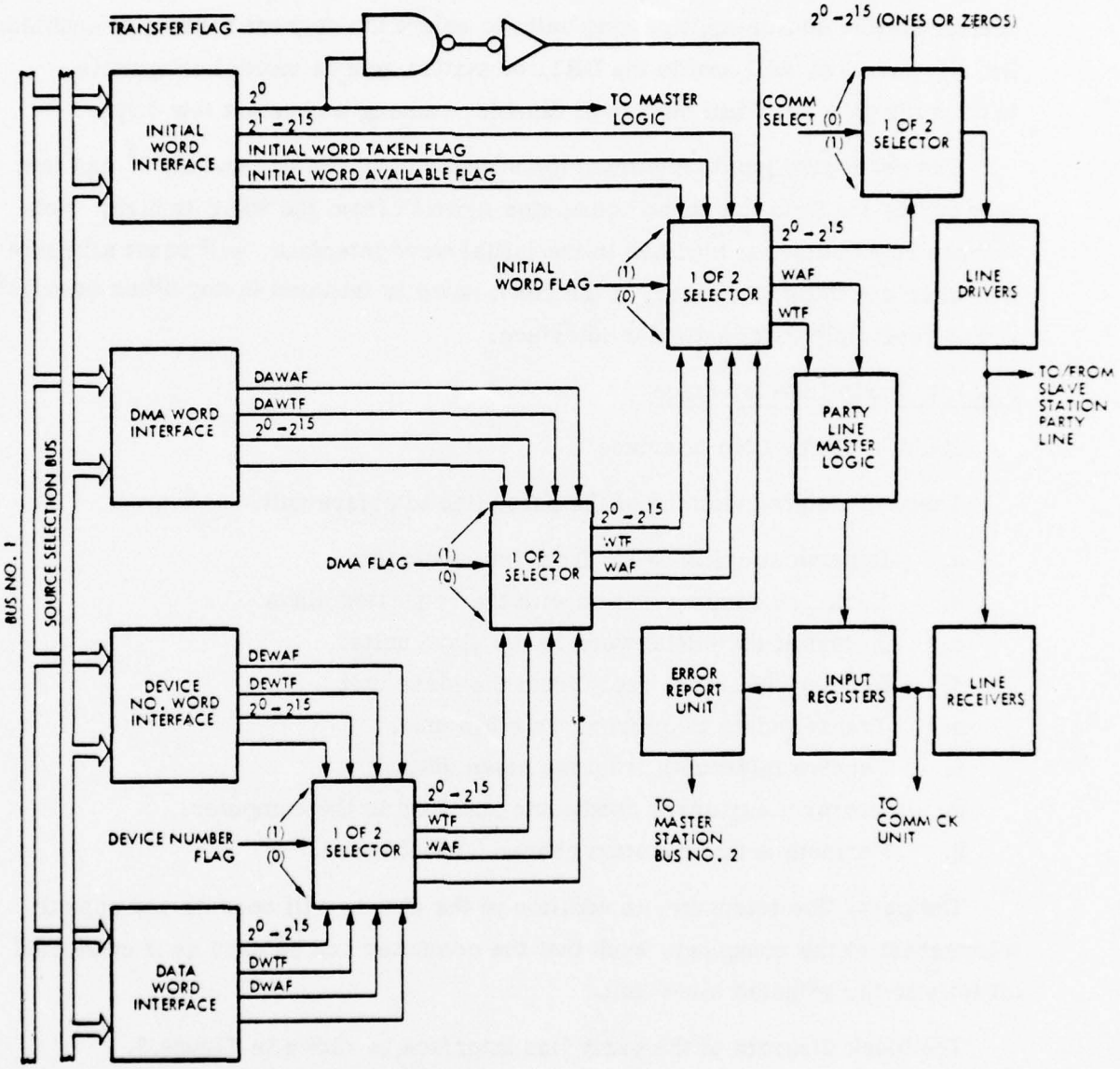


Figure 5. Party Line Interface Block Diagram

2.1.4.5 Party Line Master Logic

a. initial word from computer,

b. request for interrupt from a slave.

The interpretation of the initial word op-codes in the party line master logic is slightly different from that presented in paragraph 2.1.3. There are two differences.

a. A slave address of zero is interpreted as a system master clear.

b. Bit zero and bit 2 (transfer) are used to determine if DMA is to be used.

In the event of an address zero, only the master clear is activated and no communication to a specific slave is initiated. If bit 2 is set, indicating a transfer of data to/from a slave, bit zero is tested to determine if the transfer is to be either DMA or register. This information (bit 0) is then stripped as the slave unit performs the same in either case, and in addition bit 0 has a different meaning to a slave unit.

In the event of a detected error either

a. by the master station or a slave station initial reply, or

b. when interrupting,

the error report unit converts the bit pattern presented to the computer from that given in paragraph 2.1.3, to the pattern given in Table 7.

Flow charts of the party line master logic are shown in Figures 6 thru 10, and a glossary of terms is given in Table 8.

TABLE 7. ERROR REPORT UNIT OUTPUT

Bit	
0	General bit 1 = error found
1	Bad address compare (master station detected)
2	No initial reply (master station detected)
3	Response clock overtime (master station detected)
4	Interrupt clock overtime (master station detected)
5	Could not complete instruction (slave unit detected)
6	Local error detected (slave unit detected)
7	Normal slave completion or acknowledge bit (Except during communication checks, this bit becomes the Party Line Master Logic error report bit.)
8-15	Slave address (Will come from slave unit answering, unless bit 0 is set; then, address comes from the party line master logic stored address.)

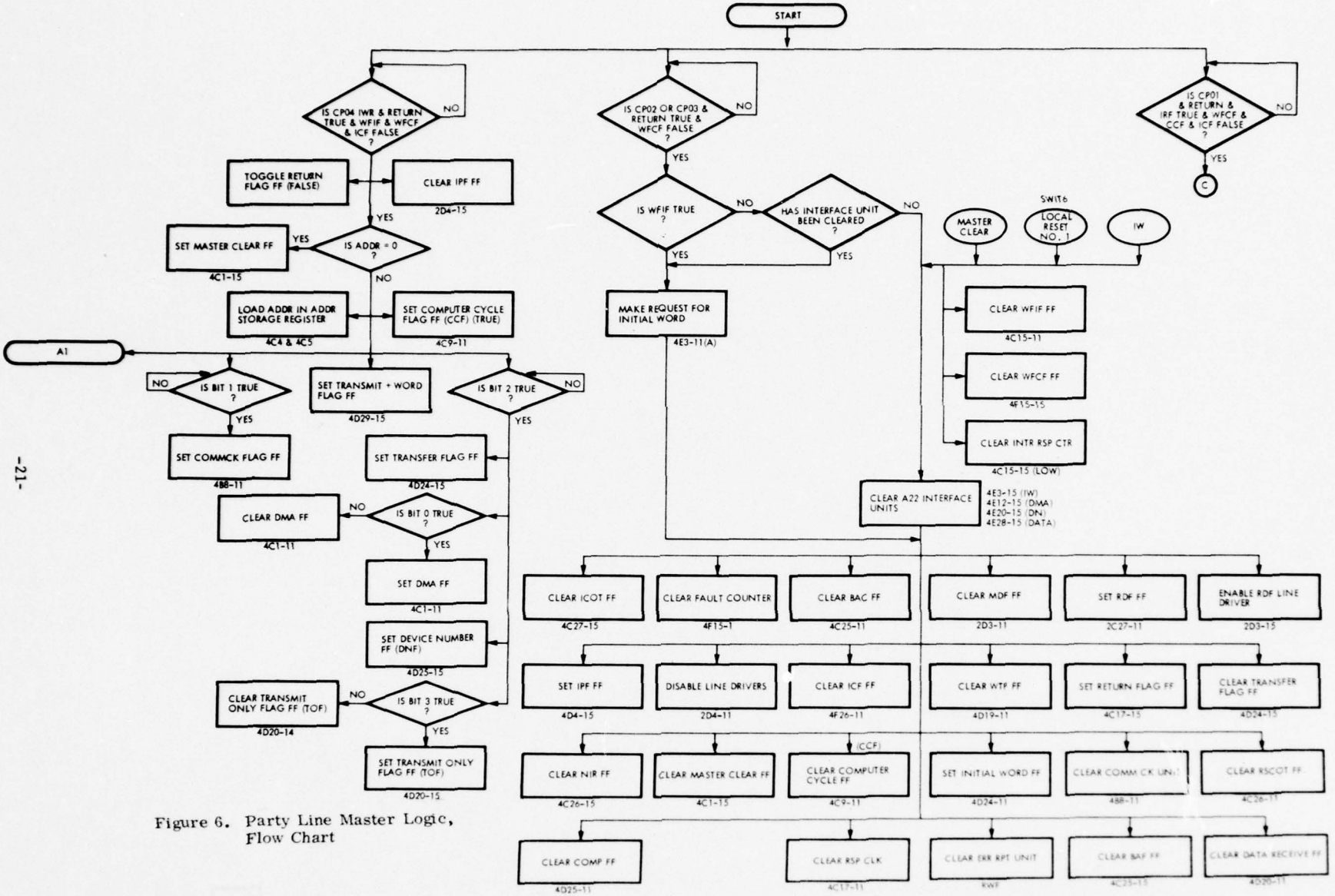


Figure 6. Party Line Master Logic, Flow Chart

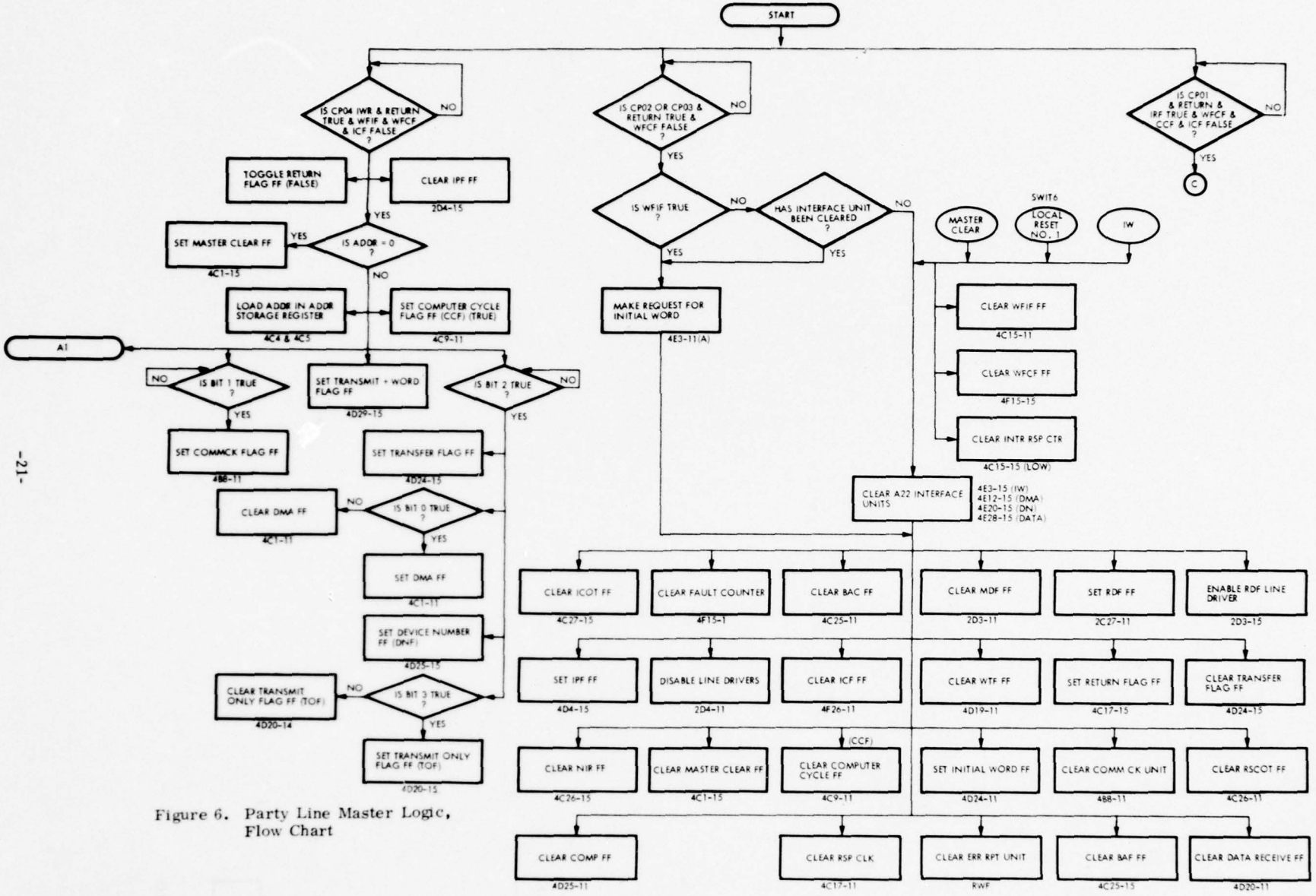
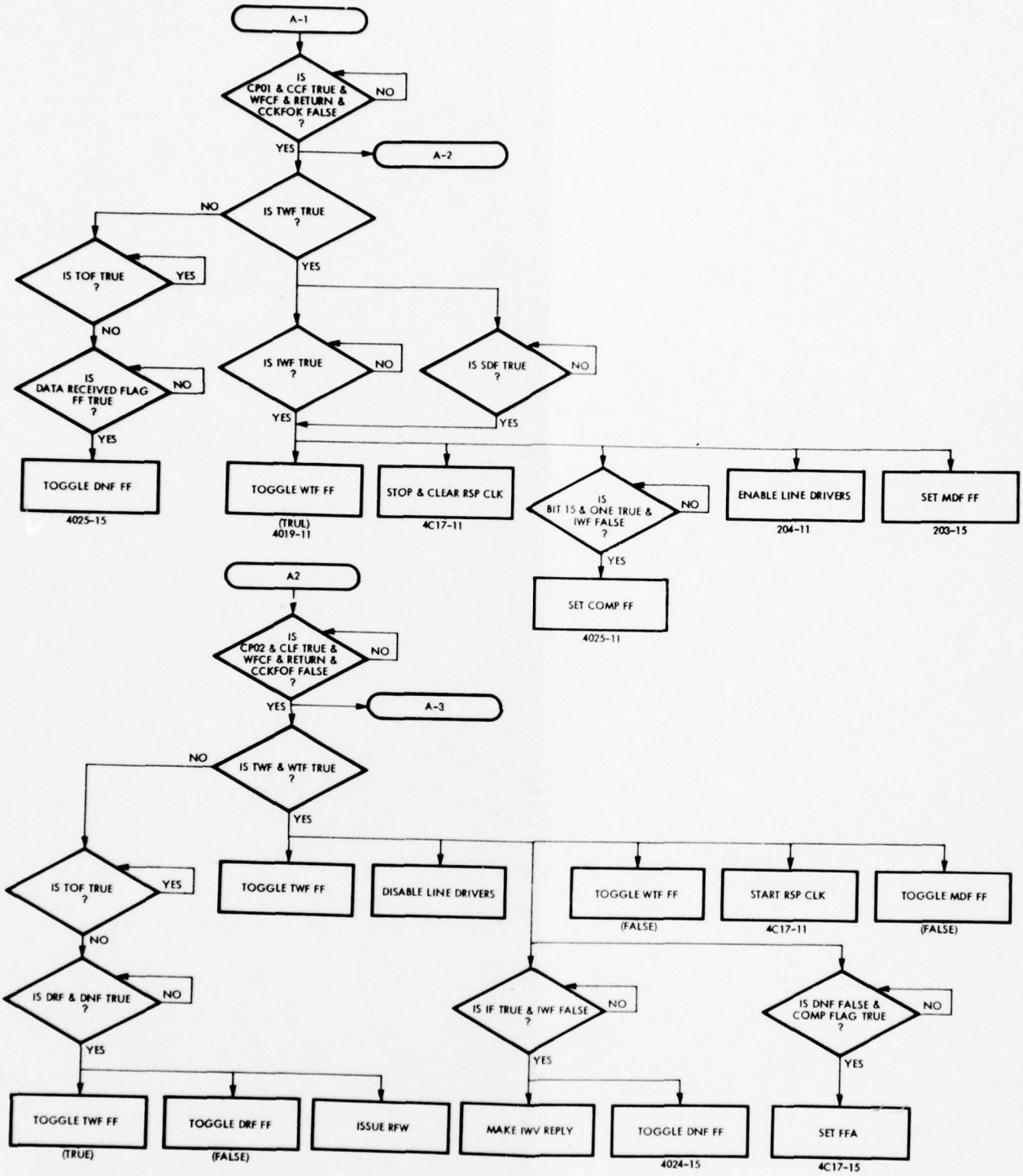


Figure 6. Party Line Master Logic, Flow Chart


A-1
IS CP01 & CCF TRUE & WFCF & RETURN & CCKFOK FALSE ?
NO
YES
A-2
IS TWF TRUE ?
NO
YES
IS TOF TRUE ?
YES
NO
IS DATA RECEIVED FLAG FF TRUE ?
NO
YES
TOGGLE DNF FF
4025-15
IS IWF TRUE ?
NO
YES
IS SDF TRUE ?
NO
YES
TOGGLE WTF FF
(TRUL)
4019-11
STOP & CLEAR RSP CLK
4C17-11
IS BIT 15 & ONE TRUE & IWF FALSE ?
NO
YES
SET COMP FF
4025-11
ENABLE LINE DRIVERS
204-11
SET MDF FF
203-15
A2
IS CP02 & CLF TRUE & WFCF & RETURN & CCKFOF FALSE ?
NO
YES
A-3
IS TWF & WTF TRUE ?
NO
YES
IS TOF TRUE ?
YES
NO
IS DRF & DNF TRUE ?
NO
YES
TOGGLE TWF FF
(TRUE)
TOGGLE DRF FF
(FALSE)
ISSUE RFW
TOGGLE TWF FF
DISABLE LINE DRIVERS
TOGGLE WTF FF
(FALSE)
START RSP CLK
4C17-11
TOGGLE MDF FF
(FALSE)
IS IF TRUE & IWF FALSE ?
NO
YES
MAKE IWV REPLY
TOGGLE DNF FF
4024-15
IS DNF FALSE & COMP FLAG TRUE ?
NO
YES
SET FFA
4C17-15

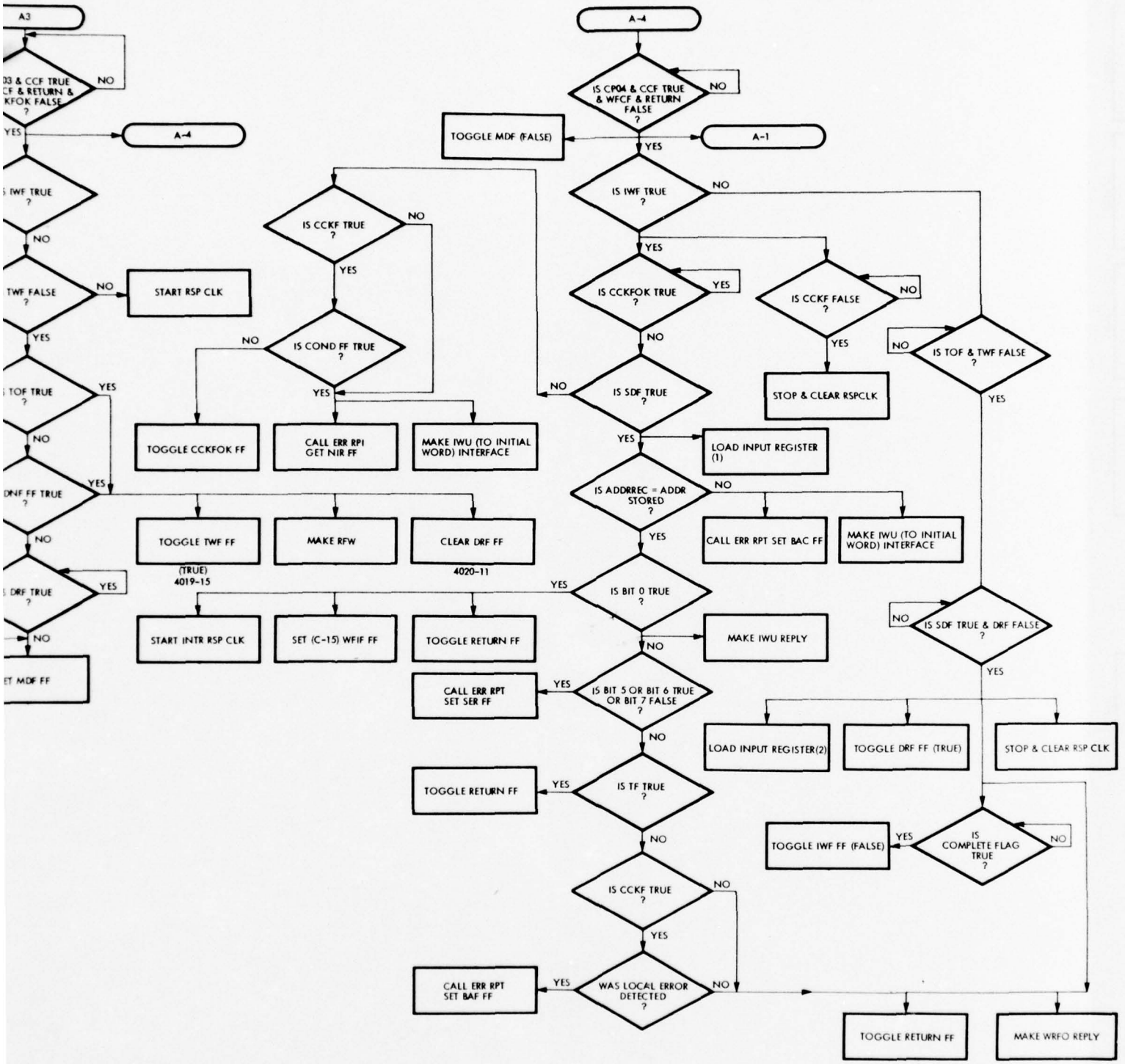


Figure 7. Party Line Master Logic (Initial Word and Data Transfer), Flow Chart

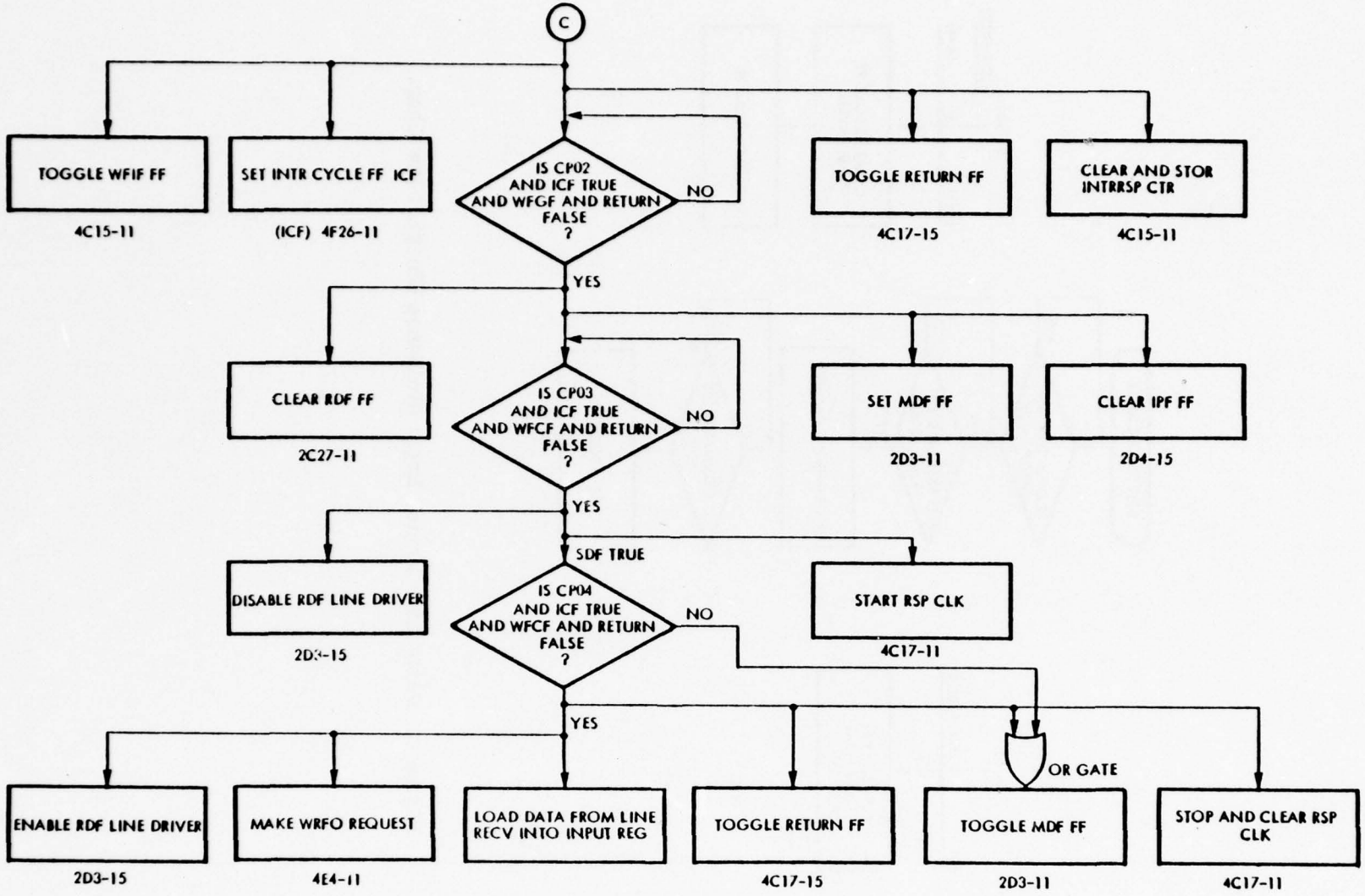


Figure 8. Party Line Master Logic (Interrupt Handling), Flow Chart

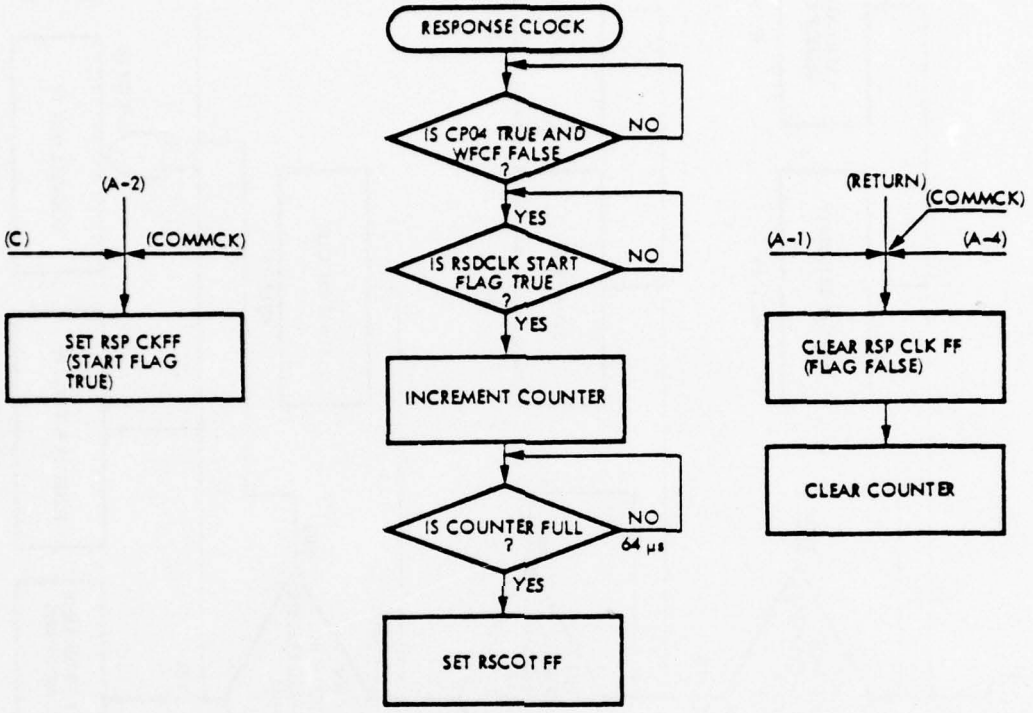


Figure 9. Party Line Master Logic (Response Clock), Flow Chart

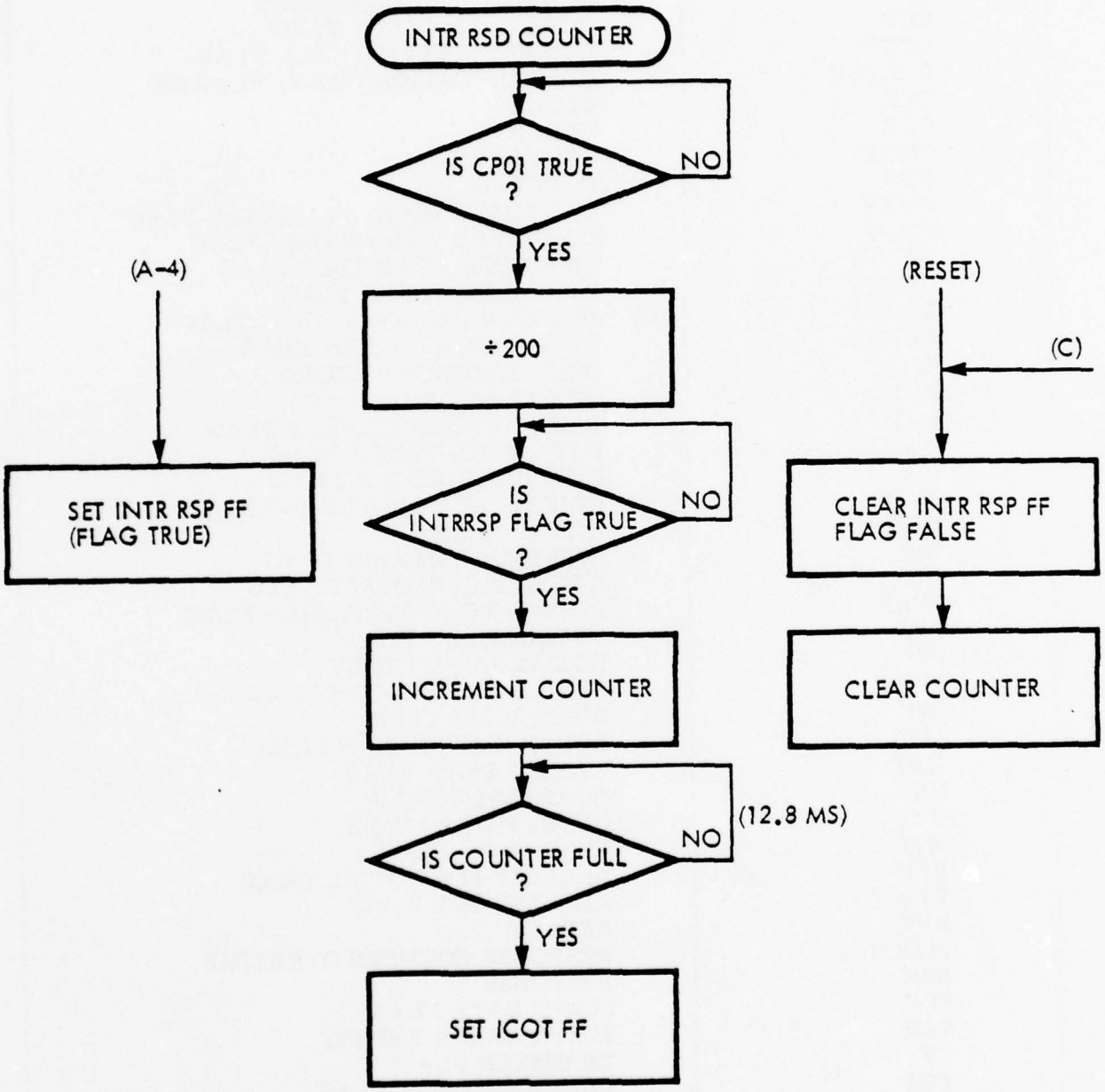


Figure 10. Party Line Master Logic (Interrupt Response Counter), Flow Chart

TABLE 8. GLOSSARY OF TERMS USED IN FLOW CHARTS

ADDR	ADDRESS
BAC	BAD ADDRESS COMPARE
BAF	BAD ACKNOWLEDGE FLAG
CCF	COMPUTER CYCLE FLAG
CCKF	COMMUNICATION CHECK FLAG
CCKFOK	COMMUNICATION CHECK FLAG OK
CLK	CLOCK
CLR	CLEAR
COMP	COMPLETE
CTR	COUNTER
CWAF	COMPUTER WORD AVAILABLE FLAG
CWTF	COMPUTER WORD TAKEN FLAG
DAWAF	DMA WORD AVAILABLE FLAG
DAWTF	DMA WORD TAKEN FLAG
DEWAF	DEVICE WORD AVAILABLE FLAG
DEWTF	DEVICE WORD TAKEN FLAG
DMA	DIRECT MEMORY ACCESS
DNF	DEVICE NUMBER FLAG
DWAF	DATA WORD AVAILABLE FLAG
DWTF	DATA WORD TAKEN FLAG
ICF	INTERRUPT CYCLE FLAG
ICOT	INTERRUPT COUNTER OVERTIME
INSTR	INSTRUCTION
IPF	INTERRUPT PERMIT FLAG
IRF	INTERRUPT REQUEST FLAG
IWAF	INITIAL WORD AVAILABLE FLAG
IWF	INITIAL WORD FLAG
IWR	INITIAL WORD READY
IWTF	INITIAL WORD TAKEN FLAG
IWU	INPUT WORD USED
LIF	LOCAL INSTRUCTION FLAG
MDF	MASTER DATA FLAG
NIR	NO INITIAL REPLY
OWU	OUTPUT WORD USED
RDF	READ DATA FLAG
RFIW	REQUEST FOR INITIAL WORD
RFW	REQUEST FOR WORD
RPT	REPORT
RSCOT	RESPONSE COUNTER OVERTIME
RSP	RESPONSE
SDF	SLAVE DATA FLAG
SER	SLAVE ERROR REPORT
TF	TRANSFER FLAG
TOF	TRANSFER ONLY FLAG
TWF	TRANSMIT WORD FLAG
WFCF	WAIT FOR COMPUTER FLAG
WFIF	WAIT FOR INTERRUPT FLAG
WRFO	WORD READY FOR OUTPUT
WTF	WORD TRANSMITTED FLAG

2.1.5 Slave Station Units

There are 11 slave station units located throughout the beamforming system. Each unit is composed of two subunits:

a. common front end (CFE),
b. card library.

The CFE is used to interface to the party line, interpret the instruction set, and perform the indicated actions. The CFE, in turn, provides the drivers and receivers for communication to the card library. These lines consist of

a. 16 bits of input data,
b. 16 bits of output data,
c. device number decoding (0-99),
d. a loading strobe.

The card library is used to hold the various printed circuit (PC) relay driver boards or special PC boards that, in turn, determine the ultimate use of the slave unit.

The three driver boards are the following.

a. In-house relay Drivers (Type A).
b. Field logic relay drivers (Type B).
c. Synthesizer drivers (Type C).

The slave station block diagram is shown in Figure 11.

2.1.5.1 Common Front End (CFE)

The CFE of each slave unit is used to interface to the party line, interpret the instructions, and initiate the desired action. In addition, the CFE contains interrupt circuits and timers.

The only interrupts are either local errors detected or requested. The requested interrupts are initiated by the use of a 10-ms timer. The 10-ms period was selected since most operations will be relay operations; this period was deemed sufficient for the relay to have achieved the desired position and for contact bounce to have ceased.

The CFE data flow is shown in Figure 12.

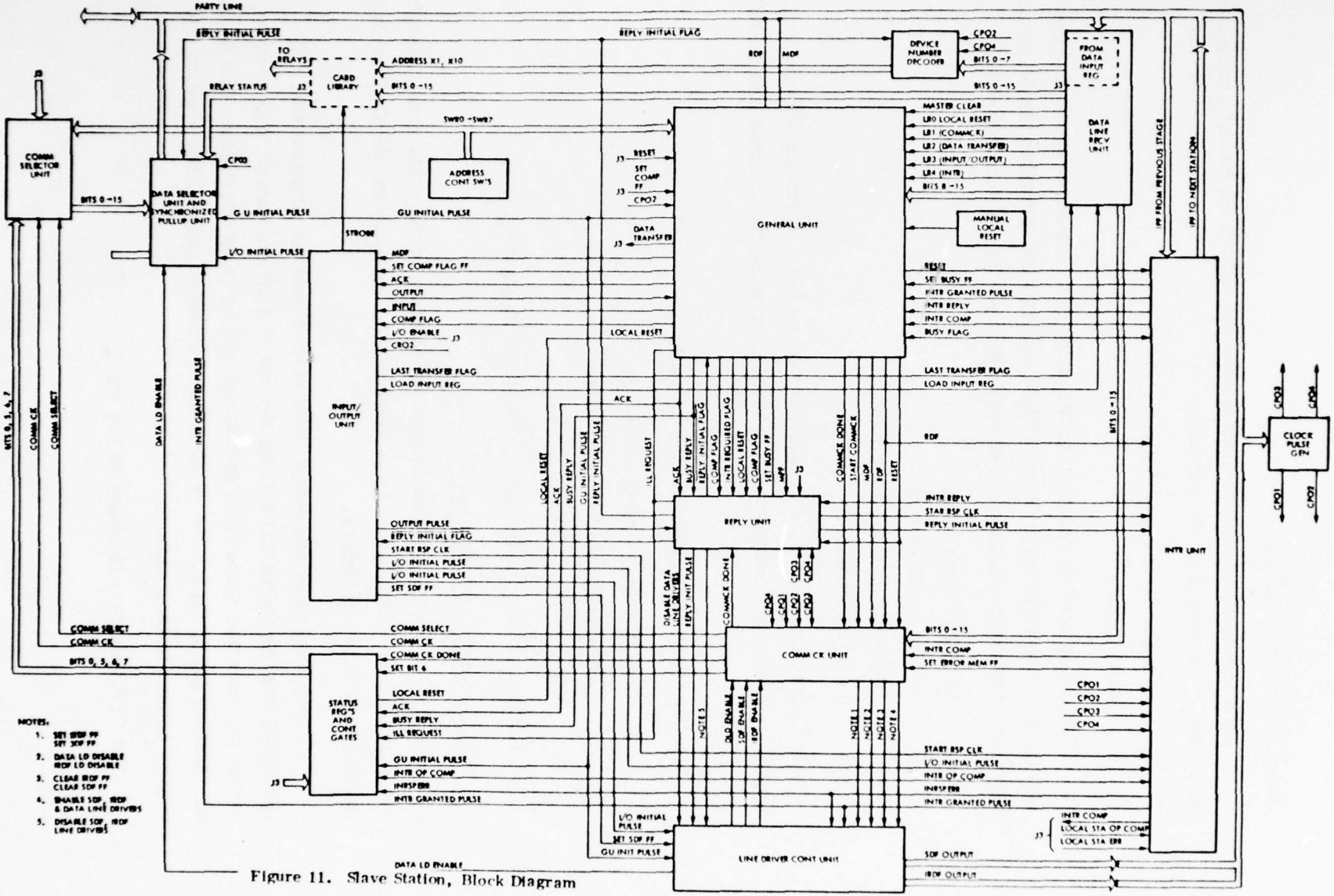


Figure 11. Slave Station, Block Diagram

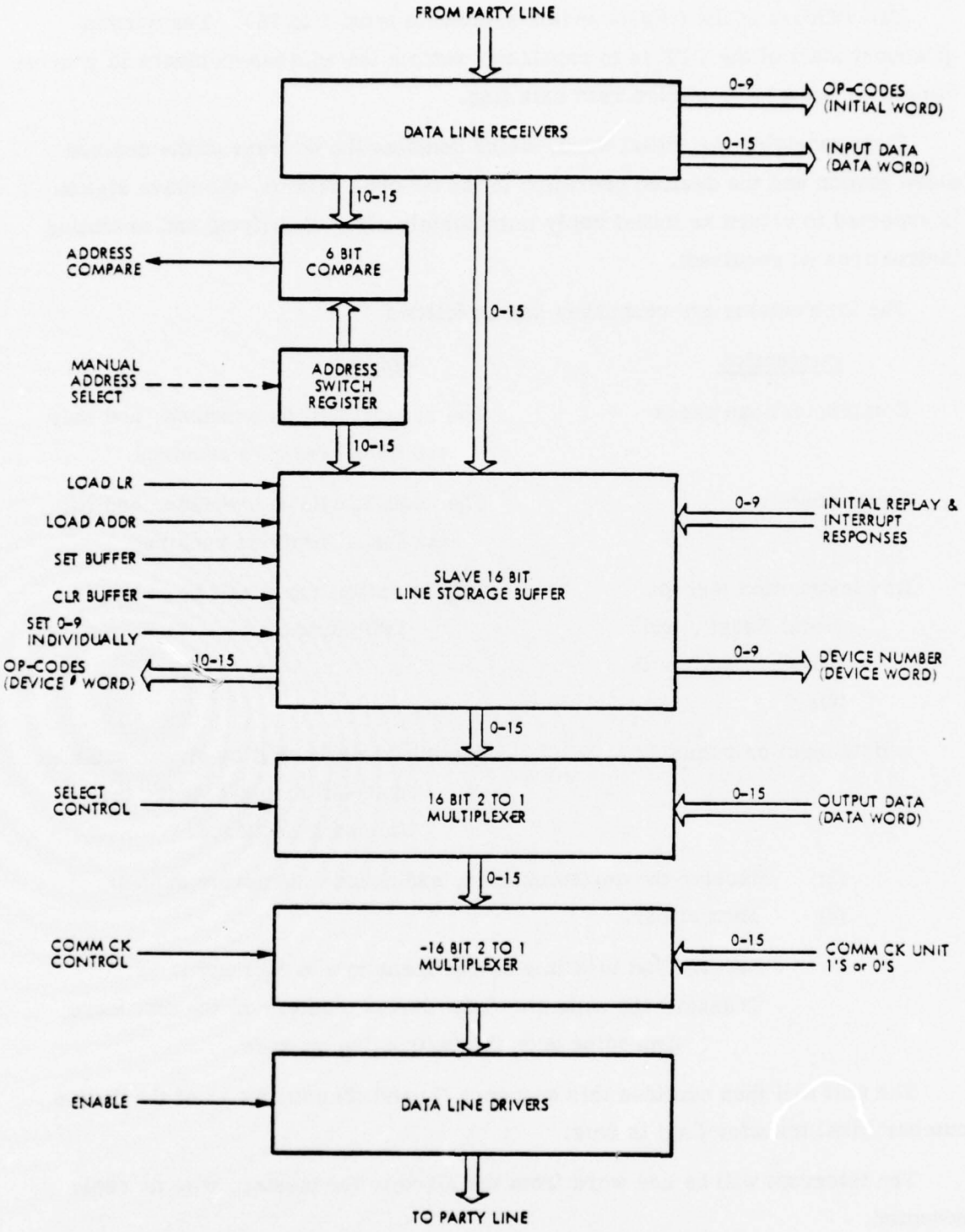


Figure 12. Slave Station, Data Flow

The address of the CFE is switch selectable from 1 to 255. The normal quiescent state of the CFE is to constantly sample the address-compare in conjunction with the master station read data flag.

Upon receiving the initial word, which contains the address of the desired slave station and the desired operation in the op-code section, the slave station is expected to return an initial reply immediately after identifying and executing instructions (if required).

The instructions are responses are as follows.

Instruction	Action
Communications check	The request will be executed, and only the initial reply is required.
Local reset	The request will be executed, and only the initial reply is required.
Any instruction (except Local Reset), and the slave station is busy	Only an initial reply will be given, indicating busy.
A data input or output	The initial reply will be given, and the unit will then set up for the following sequence.

(1) Receive the device number, and decode the device number.

(2) Then either,

- Receive and load into device memory the data word, or
- Transmit the contents of the device memory as the data word, depending upon the instruction op-code.

The unit will then continue this sequence (1) and (2) until bit 15 of the device number (final transfer flag) is true.

The interrupt will be one word from the slave to the master, with no reply accepted.

The CFE is composed of the following major sub-units.

a. General unit.
b. Reply unit.
c. Input/output unit.
d. Communication check unit.
e. Interrupt unit.

The following descriptions (paragraphs 2.1.5.1.1 through 2.1.5.1.5) are the logic sequence used by each unit to accomplish the task assigned. A glossary of terms used in the CFE is given in Table 9.

TABLE 9. GLOSSARY OF TERMS USED IN SLAVE UNITS

ACK	ACKNOWLEDGE
ALLDEN	ALL LINE DRIVERS ENABLED
CK	CHECK
CLK&CK	CLOCK
COMM	COMMUNICATION
COMP	COMPLETE
$DS2^N$	DATA SELECT 2^N
FF	FLIP FLOP
GU	GNER
ILL	ILLEGAL
IN	INPUT
INRSPERR	INTERRUPT RESPONSE ERROR
INTR	INTERRUPT
I/O	INPUT/OUTPUT
IPF	INTERRUPT PERMIT FLAG
IRDF	INTERNAL READ DATA FLAG
LD	LINE DRIVER
LOC	LOCAL
MDF	MASTER DATA FLAG
MEM	MEMORY
OP&OPR	OPERATION
$R2^N$	RELAY 2^N
RDF	READ DATA FLAG
REG	REGISTER
REQ	REQUEST
RSP	RESPONSE
RSPERR	RESPONSE ERROR
SDF	SLAVE DATA FLAG
STA	STATION

2.1.5.1.1 General Unit

Step I

IS (A) Address correct?

(B) CPO2 TRUE?

(C) Master data flag (MDF) TRUE?

(D) Read data flag (RDF) TRUE?

(E) Reply initial flag TRUE?

IF: TRUE - Continue at II.

NOT TRUE - Return to I.

Step II

(A) CLEAR complete flag.

(B) Enable SDF line driver.

(C) Load bit 6 from error memory.

(D) CLEAR SDF F.F.

(E) CLEAR IRDF F.F.

(F) Switch data selector to output address and status.

Step III

IS local reset desired?

IF: TRUE - exit to local reset unit.

NOT TRUE - Continue at IV.

Step IV

IS slave station busy?

IF: TRUE - exit to Busy Reply Unit.

NOT TRUE - Continue at V.

Step V

IS comm check desired?

IF: TRUE - Exit to comm check unit.

NOT TRUE - Continue at VI.

Step VI

IS this a data transfer?

IF: TRUE - Continue at VII.

NOT TRUE - Continue at VII.

Step VII

IS this legal for this device?

IF: TRUE - Exit to local device.

NOT TRUE - Exit to Ill. request unit.

(A) IS interrupt required?

IF: TRUE - SET interrupt required F.F.

NOT TRUE - CLEAR interrupt required F.F.

(B) IS this a legal input?

IF: TRUE - 1) Set I/O F.F.

2) Continue at VIII.

NOT TRUE - Exit to illegal request unit.

OR IS this a legal output?

IF: TRUE - 1) CLEAR I/O F.F.

2) Continue at VIII.

NOT TRUE - Exit to illegal request unit.

Step VIII

(A) SET load number F.F.

(B) CLEAR bits 0 and 5 (status register).

(C) SET bit 7 (status register).

(D) Exit to reply unit.

2.1.5.1.2 Reply Unit

Step I

(A) TOGGLE reply flag F.F.

(B) Start response clock.

Step II

IS (A) Reply flag TRUE?

(B) CP03 TRUE?

(C) Master data flag (MDF) TRUE?

IF: TRUE - Continue at IV.

NOT TRUE - Continue at III.

Step III

IS there a response error?

IF: TRUE - Exit to interrupt unit.

NOT TRUE - Return to II.

Step IV

(A) Set slave data flag (SDF) F.F.

(B) Enable data line drivers.

(C) Enable internal read data flag (IRDF) line driver.

(D) Stop and clear response clock.

Step V

IS CP04 TRUE?

IF: TRUE - Continue at VI.

NOT TRUE - Return to V.

Step VI

IS complete flag TRUE?

IF: TRUE - Continue at VII.

NOT TRUE - 1) Start response clock.

2) CLEAR reply initial flag F.F.

3) Wait for CP04 to finish.

then: 1) TOGGLE reply flag F.F.

2) Disable data line drivers.

3) Exit to I/O Unit.

Step VII

SET Reply Initial Flag F.F.

Step VIII

IS interrupt required F.F. TRUE?

IF: TRUE - Continue at IX.

NOT TRUE - Continue at XI.

Step IX

(A) SET busy F.F.

(B) Start operation complete clock.

Step X

(A) Exit to interrupt unit.

Step XI

IS CP04 finished?

IF: TRUE - Continue at XII.

NOT TRUE - Return to XI.

Step XII

(A) Disable data line drivers.

(B) Disable slave data flag (SDF) line driver.

(C) Disable internal read data flag (IRDF) line driver.

(D) Toggle reply flag F.F.

(E) Exit to general unit.

2.1.5.1.3 Input/Output Unit

Step I

IS (A) Master data flag (MDF) TRUE?

(B) CP02 TRUE?

(C) Reply initial flag TRUE?

(D) I/O enable J3 TRUE?

IF: TRUE - Continue at III.

NOT TRUE - Continue at II.

Step II

IS there a response error?

IF: TRUE - Exit to interrupt unit.

NOT TRUE - Return to I.

Step III

(A) CLEAR and stop response clock.

(B) Disable slave data flag (SDF) line driver.

(C) SET data selector to input device data.

(D) IS load number TRUE?

IF: TRUE - Load input registers (data selector).

NOT TRUE - Load device memory registers.

Step IV

IS CP02 finished?

IF: TRUE - 1) Start internal clock.

2) Continue at V.

NOT TRUE - Return to IV.

Step V

IS this last transfer?

IF: TRUE - Set Complete F.F.

NOT TRUE - No action.

AND IS this an input or output?

IF: OUTPUT - Exit to reply unit.

INPUT - Continue at VI.

Step VI

(A) Set slave data flag (SFD) F.F.

(B) Start response clock.

(C) IS complete flag TRUE AND load number flag FALSE.

IF: TRUE - Exit to reply unit.

NOT TRUE - Continue at VII.

Step VII

IS internal clock pulse finished?

IF: TRUE - 1) TOGGLE load number F.F.

2) Return to I.

NOT TRUE - Return to VII.

2.1.5.1.4 Communications Check Unit

STANDBY CONDITION

A) ALLDEN FALSE.

B) Second CP02 flag FALSE.

C) Clock input F.F. CLEAR.

Step I

A) SET Condition F.F.

B) SET Comm clock F.F.

Step II

IS CP01 finished?

IF: TRUE – 1) SET clock input F. F.
2) Continue at III.
NOT TRUE – Return to I.

Step III

IS CP02 TRUE AND ALLDEN TRUE?

IF: TRUE – 1) Disable data line drivers.
2) Disable IRDF line driver.
3) CLEAR error memory F. F.
4) Continue at IV.
NOT TRUE – Return to III.

Step IV

IS CP03P TRUE AND ALLDEN FALSE?

IF: TRUE – 1) Sample error logic.
2) SET 2nd CP02 F. F.
3) Continue at V.
NOT TRUE – Return to IV.

Step V

IS error detected?

IF: TRUE – 1) SET error memory F. F.
2) Continue at VI.
NOT TRUE – Continue at VI.

Step VI

IS CP03P finished?

IF: TRUE – 1) Toggle condition F. F. to output zeros.
2) Continue at VII.
NOT TRUE – Return to VI.

Step VII

IS CP01P TRUE AND ALLDEN FALSE?

IF: TRUE – 1) Sample error logic.
2) Continue at VIII.
NOT TRUE – Continue at VIII.

Step VIII

IS error detected?

IF: TRUE – 1) SET error memory F. F.
2) Continue at IX.
NOT TRUE – Continue at IX.

Step IX

IS CP01P finished?

IF: TRUE – 1) TOGGLE condition F. F. to output one's.
2) Continue at X.
NOT TRUE – Return to IX.

Step X

IS CP02P, 2ND CP02 flag TRUE AND ALLDEN FALSE?

IF: TRUE – 1) SET SDF F. F.
2) SET IRDF F. F.
3) Continue at XI.
NOT TRUE – Return to X.

Step XI

IS CP02P finished?

IF: TRUE – 1) Enable data line drivers.
2) Enable SDF line driver.
3) Enable IRDF line driver.
4) Continue at XII.
NOT TRUE – Return to XI.

Step XII

IS CP04P AND ALLDEN TRUE?

IF: TRUE – 1) CLEAR SDF F.F.
2) CLEAR IRDF F.F.
3) CLEAR condition F.F. to output zero's.
4) Continue at XIII.
NOT TRUE – Return to XII.

Step XIII

IS CP02P AND ALLDEN TRUE?

IF: TRUE – 1) SET bit 7 (status register).
2) SET complete F.F.
3) SET reply flag F.F.
4) Continue at XIV.
NOT TRUE – Return to XIII.

Step XIV

IS CP02P finished?

IF: TRUE – 1) TOGGLE second CP02 F.F. FALSE.
2) TOGGLE comm clock F.F. FALSE.
3) Continue at XV and exit to reply unit.
NOT TRUE – Return to XIV.

Step XV

IS CP03 TRUE?

IF: TRUE – 1) CLEAR clock input F.F.
2) Stop.
NOT TRUE – Return to XV.

2.1.5.1.5 Interrupt Unit

Step I

(A) IS (1) CP04 TRUE?
(2) Operation complete TRUE?
IF: TRUE – 1) SET interrupt request F. F.
2) SET Busy F. F.
3) Continue at II.
NOT TRUE – Continue at II.

AND

(B) IS (1) CP04 TRUE?
(2) Local station error TRUE?
IF: TRUE – 1) SET busy F. F.
2) SET error memory F. F.
3) SET interrupt requested F. F.
4) Continue at II.
NOT TRUE – Continue at II.

AND

(C) IS (1) CP04 TRUE?
(2) Local station operation complete?
IF: TRUE – 1) SET busy F. F.
2) SET interrupt request F. F.
3) Continue at II.
NOT TRUE – Continue at II.

AND

(D) IS (1) CP04 TRUE?
(2) Response error TRUE?
IF: TRUE – 1) Disable data line drivers.
2) Disable SDF line driver.
3) Disable IRDF line driver.
4) SET busy F. F.
5) SET error memory F. F.
6) SET interrupt request F. F.
7) Continue at II.
NOT TRUE – Continue at II.

Step II

(A) IS interrupt permit flag from previous station TRUE?

IF: TRUE – Continue to B.
NOT TRUE – Continue at IV.

(B) IS interrupt request flag TRUE?

IF: TRUE – 1) Interrupt Permit Flag becomes TRUE.
2) Continue at IV.
NOT TRUE – Continue at III.

Step III

(A) Make interrupt permit flag (IPF) to next station TRUE.

(B) Continue at V.

Step IV

Make interrupt permit flag (IPF) to next station FALSE.

Step V

IS (A) CP01 TRUE?
(B) Read data flag (RDF) TRUE?
(C) Interrupt permit flag TRUE?

IF: TRUE – 1) SET interrupt granted F.F.
2) CLEAR interrupt required F.F.
3) CLEAR IRDF F.F.
4) SET data selector to output address and status.
5) Continue at VI.
NOT TRUE – Continue at VI.

Step VI

IS CP02 TRUE?

IF: TRUE – Continue at VII.
NOT TRUE – Return to VI.

Step VII

(A) IS interrupt granted flag TRUE?

IF: TRUE – 1) SET complete F. F.

2) Start response clock

3) TOGGLE reply flag F. F.

4) Continue at VIII.

NOT TRUE – Return to I.

OR (B) IS (1) Response error TRUE?

(2) Interrupt granted flag TRUE?

IF: TRUE – 1) SET bit 5.

2) CLEAR bits 0 and 7.

3) Continue at VIII.

NOT TRUE – Return to I.

OR (C) IS (1) Operation complete flag TRUE?

(2) Interrupt granted flag TRUE?

IF: TRUE – 1) SET bit 7.

2) CLEAR bits 0 and 5.

3) Continue to VIII.

NOT TRUE – Return to I.

OR (D) IS busy flag TRUE?

IF: TRUE – 1) Start operation complete clock.

2) Return to I.

NOT TRUE – Return to I.

Step VIII

(A) Exit to reply unit, and continue at IX.

Step IX

IS (A) CP03 TRUE?

(B) Interrupt request flag TRUE?

(C) Interrupt granted flag TRUE?

IF: TRUE – 1) CLEAR interrupt request F.F.

2) CLEAR interrupt granted F.F.

3) CLEAR operate complete clock.

4) CLEAR local station error flag.

5) SET error memory F.F.

6) CLEAR busy F.F.

7) Continue at X.

NOT TRUE – Return to IX.

Step X

IS IPF from previous station TRUE?

IF: TRUE – Make IPF to next station TRUE.

NOT TRUE – Make IPF to next station FALSE.

THEN: STOP.

2.1.5.1.6 Support Units

The following three minor units complete the CFE unit.

a. Local reset

b. Illegal request

c. Busy reply

All three units exit to the reply unit as their termination. The following is the action initiated by each.

a. The Local Reset Unit

(1) Issues a reset pulse to the CFE

(2) Clears bits 0 and 5

(3) Sets bit 7 and the Complete F.F.

b. The Illegal Request Unit

(1) Sets bit 5 and the Complete F.F.

(2) Clears bits 0 and 7.

c. The Busy Reply Unit

(1) Sets the Complete F. F.

(2) Sets bit 0

(3) Clears bits 5 and 7.

2.1.5.2 Slave Card Library

The card library is designed to hold 26 printed circuit cards or 13 wire-wrapped cards. One location, Slot 14, is used as part of the CFE and contains the line drivers and device number decoding for the remaining slots.

The card library back plane supplies power to the correct pins for several voltages:

a. 5V dc

b. 28V dc

c. -12.5V dc

as well as separate grounds for each supply. The back plane distributes the device number decoding such that each slot has a permanent set of four device numbers assigned; thus, 100 (0-99) device numbers are assigned to the 25 slots.

In addition, the back plane supplies, in parallel to all device numbers,

a. 16 buts of input data,

b. 16 bits of output data, and

c. the load strobe.

With this arrangement, the function performed by the card library is determined by the type of PC card used in any given slot. There is no requirement that all the cards be of a given type.

At present, there are only three types of cards available:

a. Relay Driver Type A,

b. Relay Driver Type B, and

c. Relay Driver Type C.

2.1.5.3 Relay Driver Boards

All driver boards have the following four features in common.

a. Memory storage for each device
b. Address decoding (0-3)
c. Transistor driving circuits
d. Read back circuits

All memory storage input is in parallel, and loading is directed by the address decoders. Each storage bit has a discrete circuit and an individual output. The readback circuits are diode OR'ed into the parallel output data lines on the back plane.

2.1.5.3.1 Type A

The Type A driver boards utilize all four device numbers and all 16 bits of data. The unit is used for driving the beamformer relays as a current sink. A typical Type A driver is shown in Figure 13.

A slave station using solely Type A drivers would be capable of controlling 1600 individual relay circuits with up to four relays per circuit.

2.1.5.3.2 Type B

The Type B driver board utilizes all four device numbers and eight bits of data. The unit is used for driving the relay logic, located at the elements in the array field, as a current source. A typical Type B driver is shown in Figure 14.

A slave station using solely Type B drivers would be capable of controlling 800 individual relay circuits with up to four relays per circuit. However, since each element has eight functions to control, the arrangement was to divide the 384 elements into four blocks of 96, which represents 24 columns of four rows each. These units were installed in slave units 7 through 10.

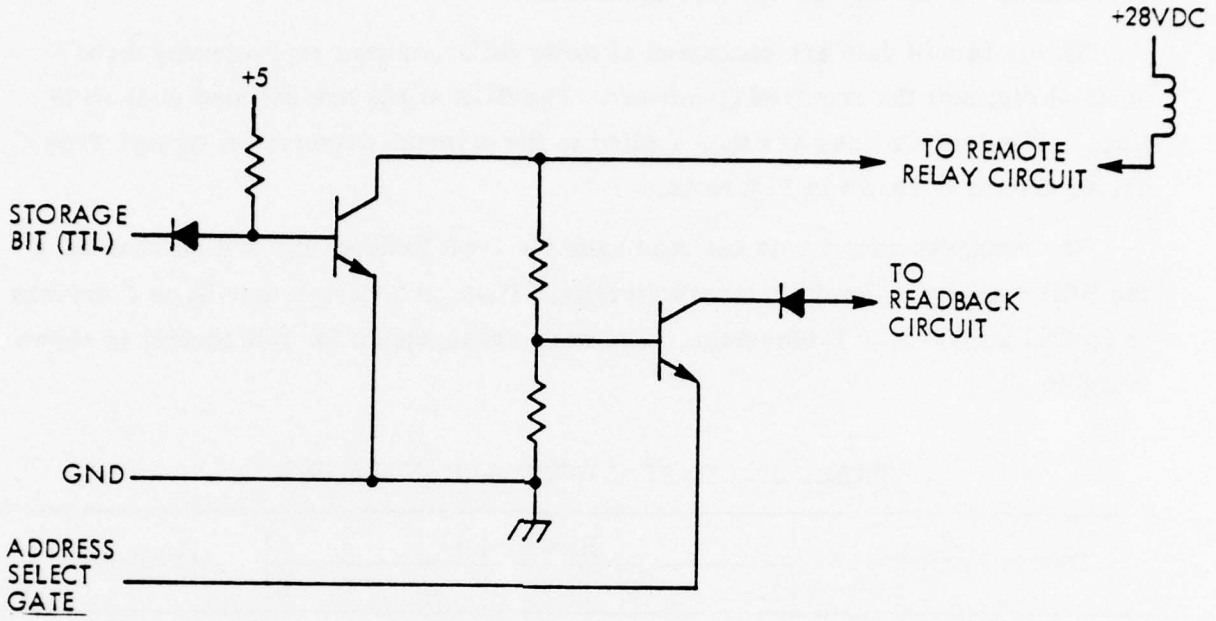


Figure 13. Typical Driver (1 of 64) Used on Relay Driver Type A

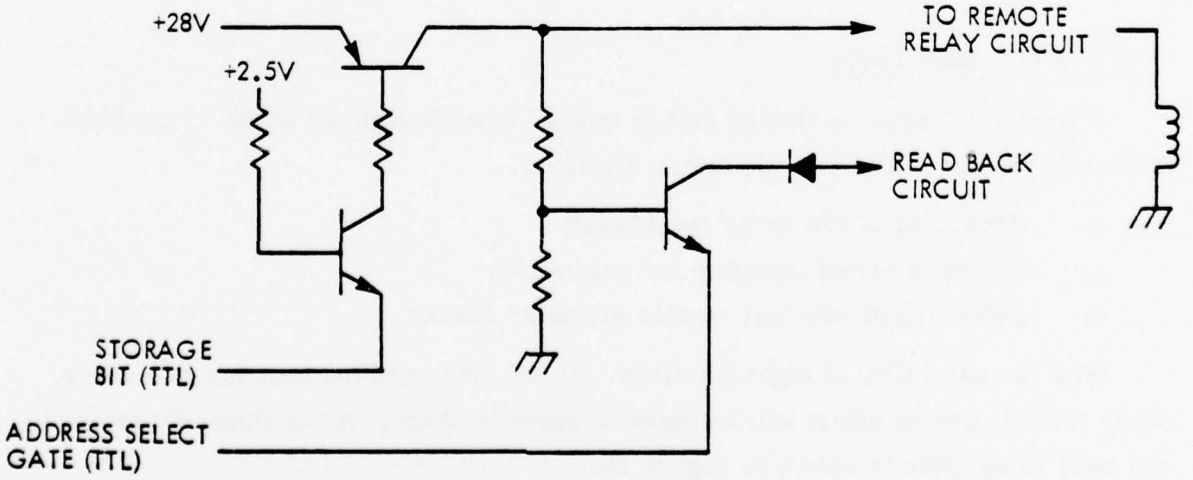


Figure 14. Typical Driver (1 of 32) Used on Relay Driver Type B

2.1.5.3.3 Type C

The Type C driver board utilizes one device number and 12 bits of data. The unit is used for driving an HP-5100 synthesizer.

The 12 bits of data are composed of three BCD numbers representing three decimal digits of the required frequency. The BCD words are decoded to 1-of-10 lines. The 1-of-10 lines are then applied to the discrete drivers. A typical Type C driver circuit is shown in Figure 15.

The readback circuits do not read back the 1-of-10 decoding, but do read back the BCD numbers from the memory storage. Thus, it requires four Type C drivers to control an HP-5100 synthesizer. The word arrangement for full control is shown in Table 10.

TABLE 10. TYPE C WORD ARRANGEMENT

Device Number	Bits (0-11)			Frequency
	11 87	43	0	
X		X10	X1	MHz
X-4	X100	X10	X1	kHz
X-8	X100	X10	X1	Hz
X-12	X.1	X.01		Hz

2.1.6 Field Relay Logic

A field relay logic switching unit is located at each element in the array field. This unit provides the following major functions.

a. Reversing of the array (east/west)
b. Selective signal injection for calibration
c. Removing of external signals for noise checks

With the capability of eight functions, the various combinations are numerous, many trivial, and no effort will be made to describe them. A functional diagram of the field relay logic is shown in Figure 16.

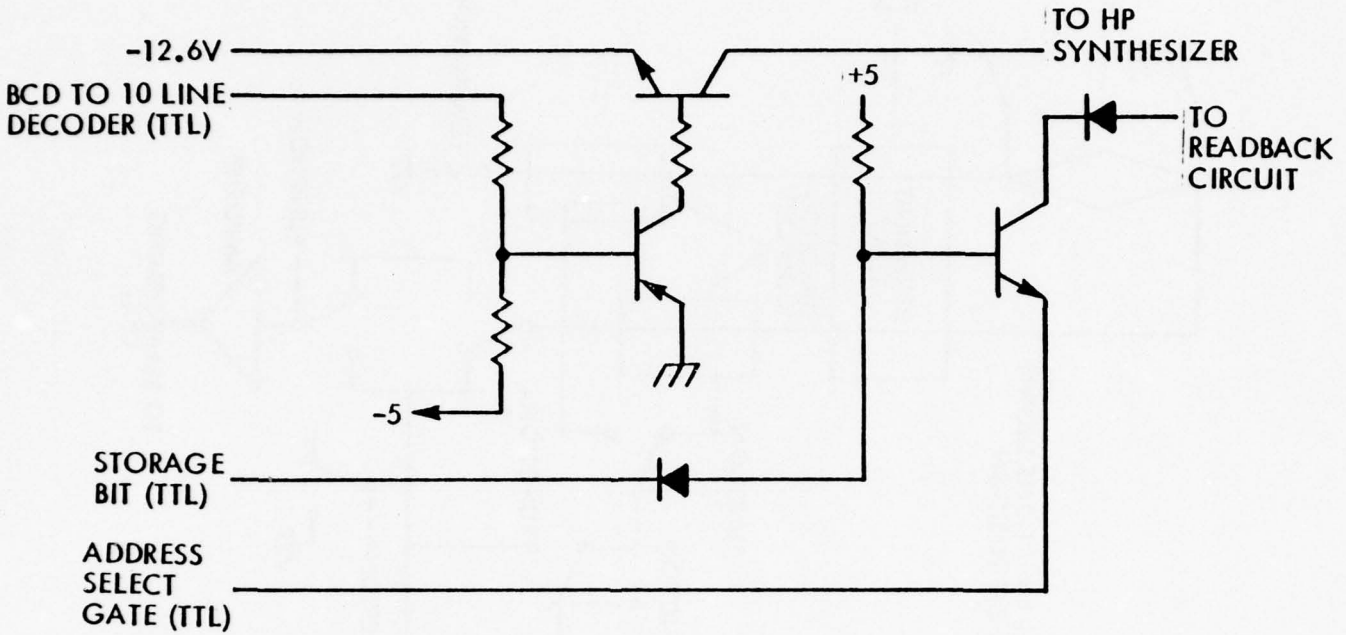


Figure 15. Typical Driver (1 of 30) Used on Relay Driver Type C

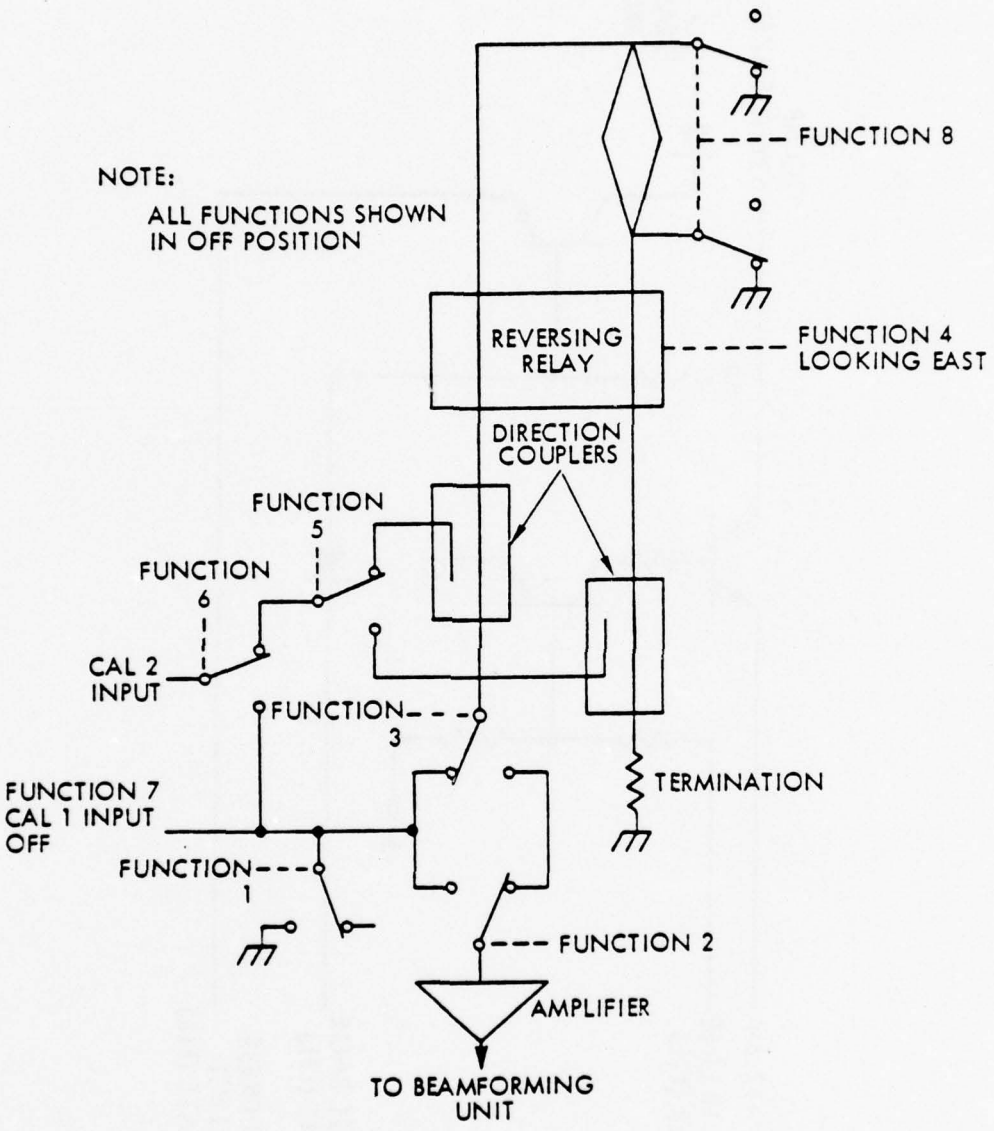


Figure 16. Field Relay Logic, Functional Diagram

2.1.7 Beamformer Trim Units

The design of the beamforming unit incorporated two basic trim units: the broadside trim and the endfire trim boards. These units utilized the same basic principal of using a DPDT relay to insert or remove delay units or attenuation units. Conceptually, there are no differences; physically, there are only three differences in the boards:

a. the number of delays and attenuation units,

b. the impedance of the circuits (endfire is 75 ohms and the broadside is 50 ohms), and

c. the granularity available.

The broadside trim boards, in order to obtain small values of attenuation changes, use an electronic attenuator followed by six bits of fixed attenuation in a binary series, starting at 2 dB through 64 dB. Thus, the electronic attenuator is to provide the 0 to 2 dB attenuation range. The block diagram of the electronic attenuator and broadside trim board is shown in Figure 17.

The electronic attenuator has the additional feature of providing a 180-degree phase shift controlled by the sign bit. This feature allows greater flexibility in generating desired aperture weighting coefficients for experimental beam shapes.

The results of the breadboard tests made on the electronic attenuator are shown in Figure 18.

The delay units on the Broadside Trim Board were to be a binary series of 16 bits, beginning with bit 0 as 0.1 inch of delay; thus, bit 15 would have been 273.067 feet for an array steering angle of ±14.5 degrees.

2.2 DIGITAL CONTROL SOFTWARE

The software delivered with the PDP 11/40 was the standard DEC support software (i.e., EDIT, PIP, MACRO, etc.), and the operating system RT11-VO2B was selected. In addition, basic and Fortran with graphics were obtained.

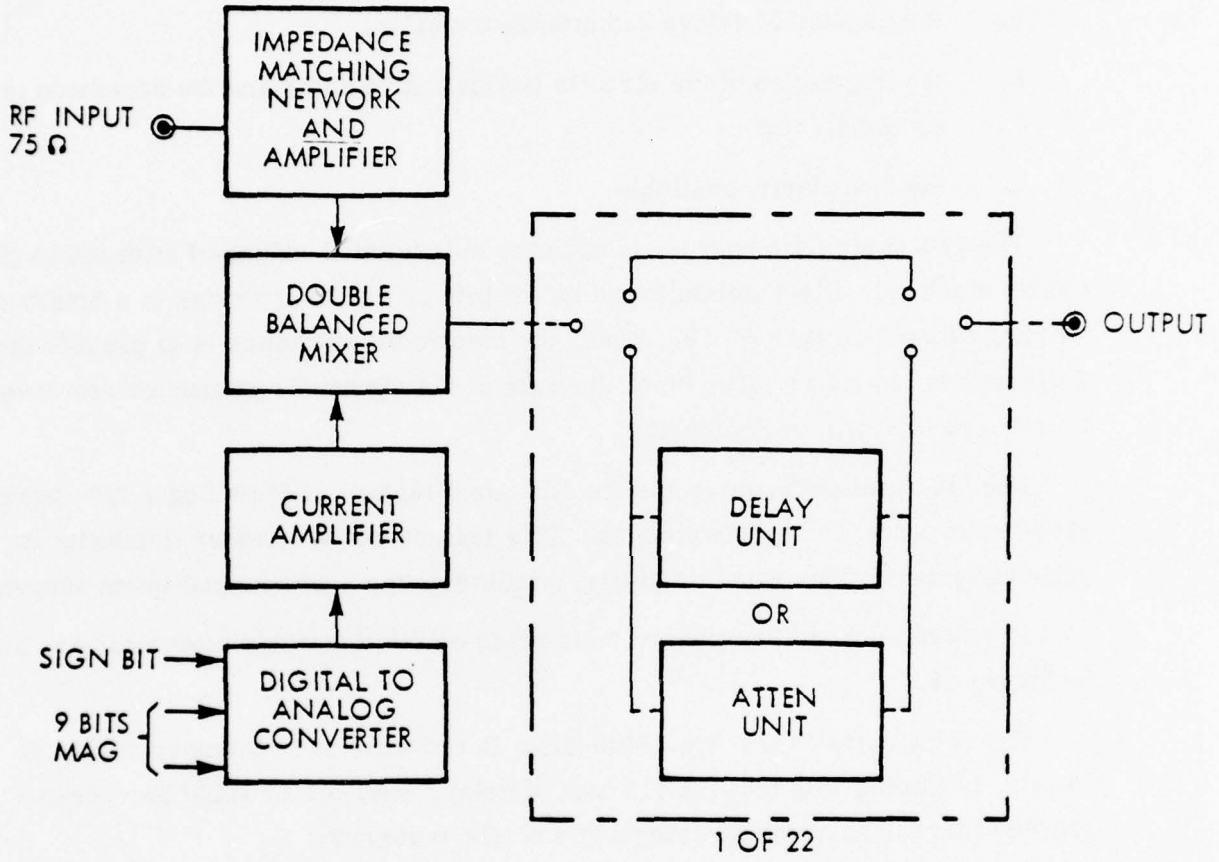


Figure 17. Electronic Attenuator and Broadside Trim Board, Block Diagram

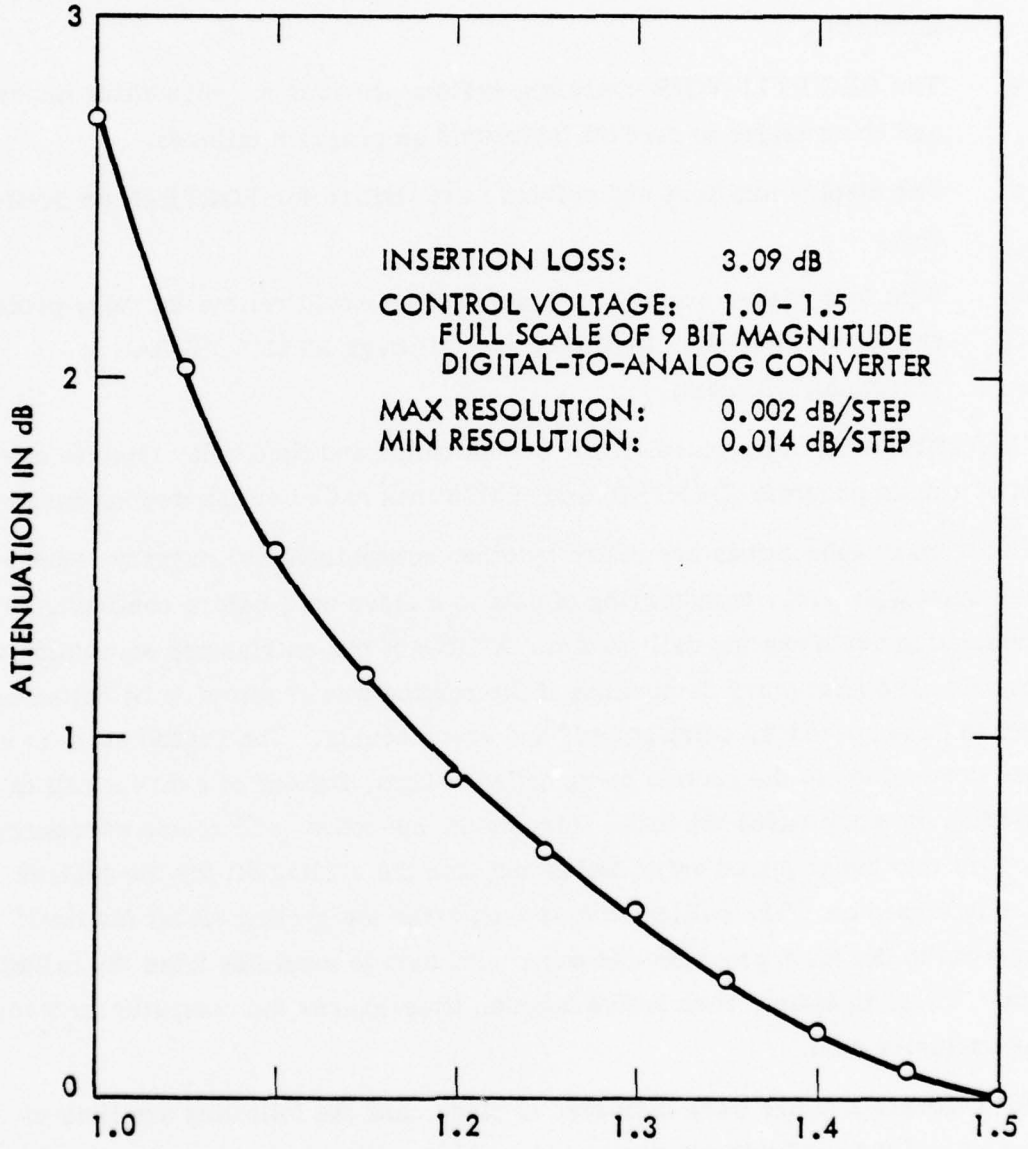


Figure 18. Electronic Attenuation as a Function of Control Voltage

For the following reasons, it was decided that all programs (including interrupt handlers) would be written in Fortran.

a. The time required to write programs would be reduced.

b. DEC FORTRAN and SYSLIB allow absolute addressing (for interrupt handlers).

c. The DEC RT11-V02B operating system provides for scheduling interrupts and abort tables to turn off interrupts on program failures.

d. The display handlers and software are written for FORTRAN or BASIC only.

e. With four disks, overlay program loading would relieve memory problems caused by FORTRAN inefficiencies (although RT11 FORTRAN is reasonably efficient).

The software for the control of the beamforming and field relay logic is composed of a main program (DEXTER) and subroutines called by the main program.

Since many subroutines are called by other subroutines and in many cases a routine must wait (i.e., transferring of data to a slave unit) before continuing, it was decided to use a cycling call system. A table of two entries per subroutine was generated. The first entry determines if the routine was required to be called and is used to pass op-codes, entry points, and error counts. The second entry is used to pass data values to the routine being called. Thus, instead of a direct call to a subroutine, in many cases the data, entry point, op-codes, and return addresses are loaded into the required entry tables and then the cycling bit for the desired routine is turned on. The calling routine then turns the cycling bit off for itself and returns to the main program and waits until data is available from the called routine. Thus, in effect, each active function time-shares the computer instead of having exclusive use.

All software listings were delivered in place, and the following are limited discussions of the available routine.

2.2.1 Main Program (DEXTER)

The main program DEXTER performs the following.

a. Provides the common storage areas (in a root section for overlay loading) and initializes the common with system constants.

b. Identifies the interrupt handler routines and provides abort tables to the RT11-VO2B operating system.

c. Calls the routines for the operator to determine the beamformer initialization parameters.

d. Checks the master station for proper operation.

e. Checks each slave unit for proper operation.

f. Initializes the beamformer according to the operator requirements and checks through readback that all units were set.

g. Starts cycling through the functional subroutines and those requested.

2.2.2 Subroutines

The subroutines called directly, or indirectly by the main program, can be separated into the following seven major categories which are discussed in Paragraphs 2.2.2.1 through 2.2.2.7.

a. Master Station Control

(1) Master Station Linkage (MSTLNK)
(2) Master Station Linkage Control (MLKCON)
(3) Master Clear (MSTCLR)

b. Communications to Slave Units

(1) Register Transfer (REGTRN)
(2) Direct Memory Access Transfer (DMATRN)
(3) Send Initial Word (SENDIW)
(4) Send Device Number (SENDDN)
(5) Send and Receive Data (SRDATA)
(6) Read Master Station (REMSTR)
(7) Direct Memory Access Interrupt (DMAINT)

c. Slave Station Control

(1) Local Clear (LCLCLR)
(2) Manual Local Clear (MANCLR)
(3) Communication Check (COMMCK)
(4) Manual Communication Check (MANCMK)

d. System Initialization

(1) New Start (NSTART)
(2) Field Function Control (FFUCON)
(3) Display Control (DPYCON)
(4) Roll Call (ROLCAL)
(5) Initialize Field (INTFLD)

e. System Control

(1) Cycle Control (CYCCTL)
(2) Synthesizer Tuning (SYNTUN)
(3) Get Synthesizer Frequency (GFREQ)

f. Diagnostics and Maintenance

(1) Element Function Control (EFUCON)
(2) Element Initialization (INTELM)
(3) Relay Driver Board Tests (NSTEP)
(4) Slave Unit Bit Test (BITTST)

g. Ancillary

(1) Device Number (Integer) to BCD (DEVBCD)
(2) Field Logic Bit Configuration Decoding (FLDDEC)
(3) Restart Timing (TIMER1)
(4) Error Report Decoding (ERRRPT)
(5) Beamformer (Broadside) Logic Bit Configuration Decoding (HSEOUT)

2.2.2.1 Master Station Control

The subroutines in this category control operations within the Master Station only (except master clear, which will also clear all slave units).

2.2.2.1.1 Master Station Linkage (MSTLNK)

Subroutine MSTLNK is used to make the source selection (or linkage) of the DR11 interface units and the master station interface units on a one-to-one pairing.

There are five formal calling parameters.

a. DR11PN = the DR11 to receive the source selection

b. MSUNIT = master station to receive source selection

c. LINKCD 0/ENABLE, 1/DISABLE

d. MSTRST 0/NO RESET, 1/RESET

e. DR11LR = DR11 to be used in linking (1-4)
(A 0 value causes DR11PN to be used.)

The DR11 addresses are given in Table 11.

TABLE 11. DR11 ADDRESSES

DR11	C1	C2	C3	C4
Status	167770	167550	167740	167760
Output Buffer	167772	167752	167742	167762
Input Buffer	167774	167754	167744	167764
Vector	310	350	360	340

DR11	B5	B6
Word Count	172410	172430
Block Address	172412	172432
Status	172414	172434
Data	172416	172436
Vector	124	300

2.2.2.1.2 Master Station Linkage Control (MLKCON)

This subroutine recalls a display file (see Paragraph 2.3.2) and pictorially displays a drawing similar to Figure 4 on the GT44 graphic display. When the operator selects a DR11 and master station pair, the program then displays the connection as a line between the two blocks representing the selected pair. An example of the display is shown in Figure 19.

The linkages so selected are stored in the arrays DR11LK and MSUNLK and are recorded on disk (RKO) in data file STRVAL.DAT for future use on power-up starts. The routine then returns to the main program, which in turn, calls MSTLNK to make the selected linkages.

2.2.2.1.3 Master Clear (MSTCLR)

This subroutine has seven functions.

a. Checks to determine if any transfers are in progress.

b. If a transfer is in progress, clears party line interface to terminate transfer.

c. Checks to determine if a slave is interrupting.

d. If interrupt is available, inputs interrupt word.

e. Checks master station to determine if clock pulses are available (i.e., switch left in manual).

f. Causes a system master clear.

g. Initializes all control tables to zero or inactive conditions.

There are three exits from this routine (two are fault conditions, plus the normal exit).

a. Master station malfunction (fault – STOP).

b. Slave station malfunction (fault – STOP).

c. Initialize tables and return to main program.

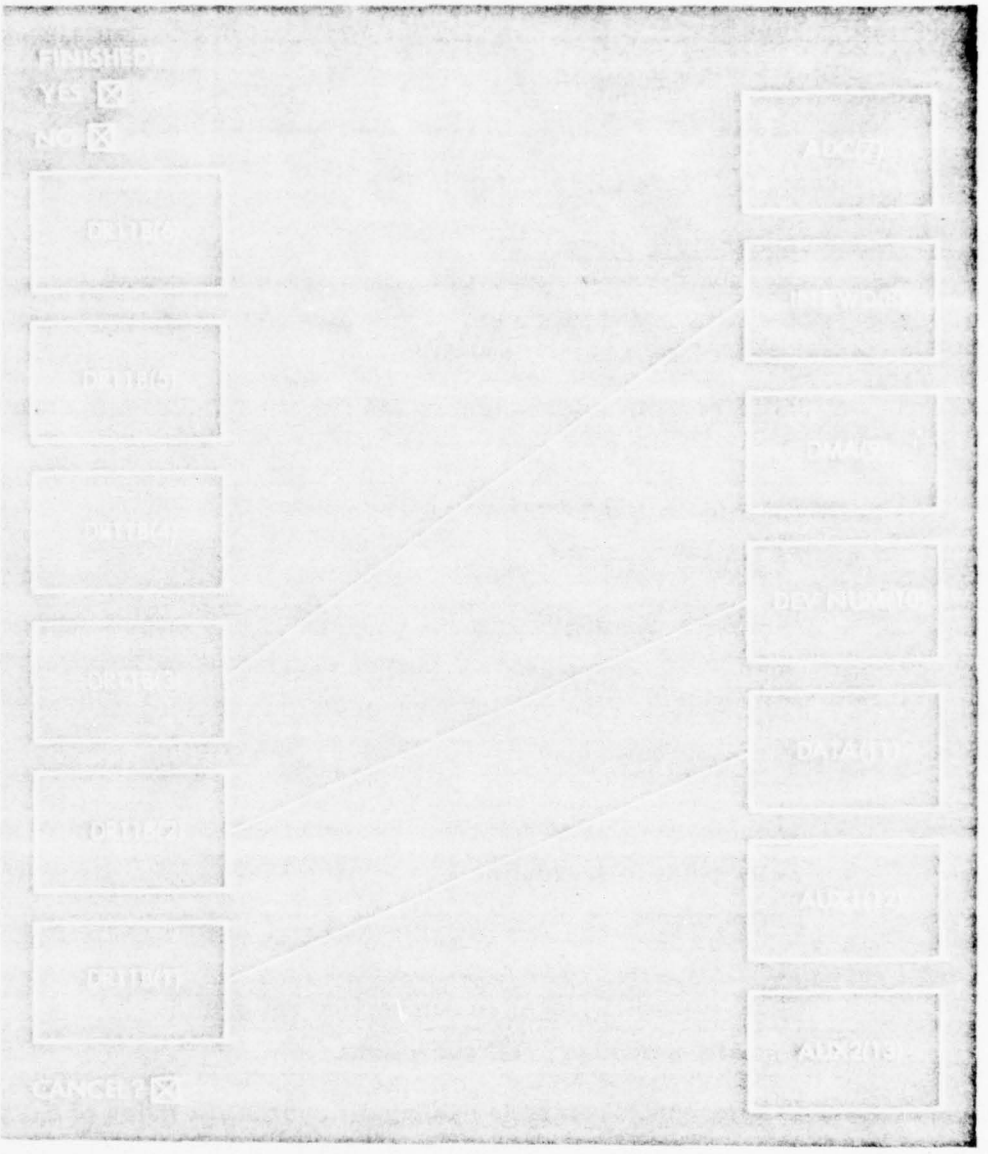


Figure 19. Graphic Display for Master Station Linkage Control (MLKCON)

The fault conditions arise from

a. attempting to get a request for initial word and failing three times, resulting in master station malfunction, and

b. receiving three consecutive interrupts from slave units, after master clearing system, resulting in slave station malfunction.

The flow chart for MSTCLR is shown in Figure 20.

2.2.2.2 Communications to Slave Units

The following group of subroutines are used to transfer data to or from a slave unit or to control commands to a slave unit.

To control the transfer of information to and from the slave units, two tables are involved:

a. cycle control table (CYCTBL), and
b. transfer table (TRNTBL).

The CYCTBL entries (nominally two per subroutine) are used to control the time-sharing or cycling of subroutines, pass op-codes, error counts, and data. One word is used to pass data, and the cycling control word is divided as shown (except if noted for a specific subroutine).

Bit	
0-3	Entry point in subroutine
4-6	Error counts
7-14	Op-codes
15	Cycle control 1 (negative number) do not call 0 (positive number) call subroutine

Each subroutine cycling control entry is assigned a subroutine index (ISUBID).

The transfer table (TRNTBL) is the table in which all data to or from a slave is stored; in addition, it is used to store control information. Each subroutine using this table is assigned an area and transfer base index (TXBASE), if required.

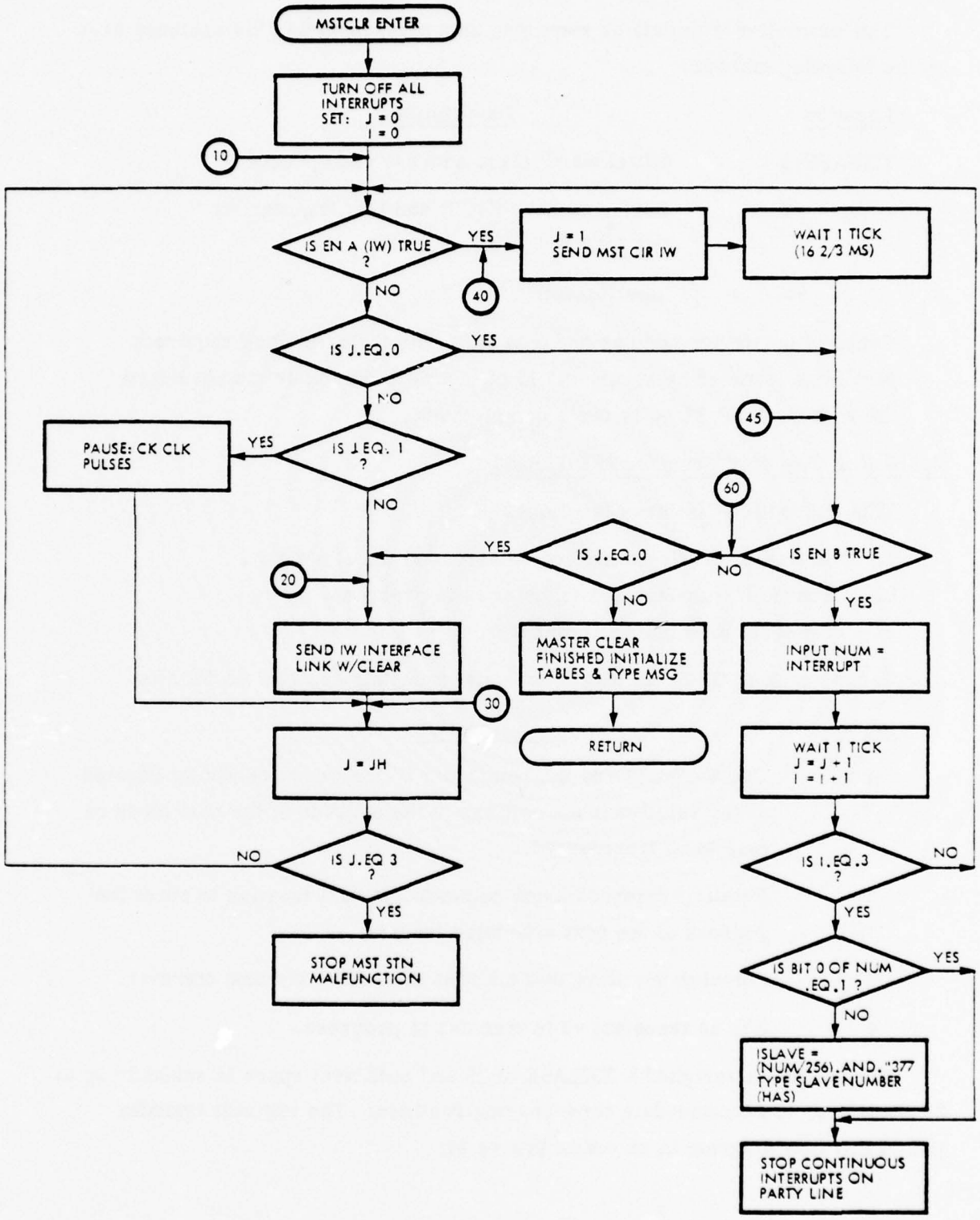


Figure 20. Master Clear (MSTCLR), Flow Diagram

The subroutine with data or requiring data must initialize the assigned area in the following manner.

Location	Description
TXBASE+1	Initial word (slave address and op-code)
+2	Device number (BCD) and last transfer bit (bit 2^{15}) with either a 1 or 0
+3	Data (in/out)

Continue the device number and data pairs for the data block required; terminate transfer by setting bit 15 of last device number (do not negate the word as PDP 11/40 is two's complement).

2.2.2.2.1 Register Transfer (REGTRN)

This subroutine has three functions.

a. Initiate register-controlled transfers to the slave unit.
b. Schedule transfers if a transfer is in progress.
c. Schedule interrupts if requested.

There are four CYCTBL locations assigned to this routine, as follows:

Index	Description
1	Cycle control and the remainder of the word are wholly devoted to the relative index pointing to the address of the next block of data to be transferred.
2	Relative-deferred index points to the next location to store the address of the next scheduled transfer.
3	Contains the slave unit address to receive the next transfer.
4	0 is no transfer; -1 is transfer in progress.

The routine is assigned a TXBASE of 19 and sufficient space to schedule up to 30 transfers, to accommodate time-sharing routines. The register transfer (REGTRN) flow diagram is shown in Figure 21.

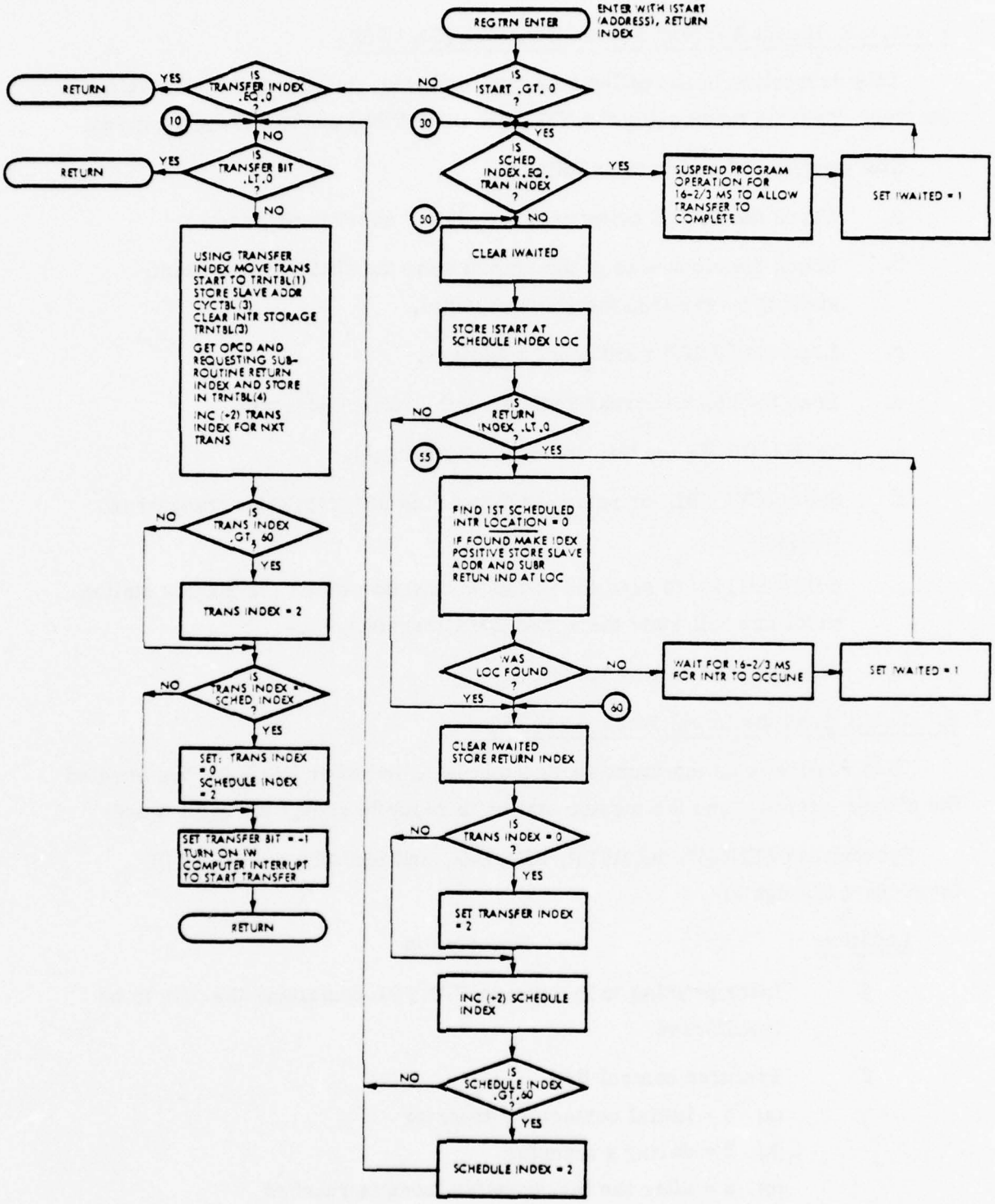


Figure 21. Register Transfer (REGTRN), Flow Diagram

2.2.2.2.2 Direct Memory Access Transfer (DMATRN)

This subroutine is not called on a cycling basis, but directly by the using routine. Entry is made using the TXBASE and ISUBID of the calling program.

The sequence of action is as follows.

a. Check the DR11B being used for status; abort if abnormal.

b. Check the block size of the transfer for the DR11B word count; abort if larger than the slave capacity.

c. Load the DR11B word count registers.

d. Load the block address register with primary location.

e. Set the DR11B GO bit, and enable interrupts.

f. Set up CYCTBL for return to the calling program after transfer is complete.

g. Call REGTRN to send the initial word (after which the master station hardware will start the actual DMA transfer).

h. Return to the calling routine.

2.2.2.2.3 Send the Initial Word (SENDIW)

This routine is an interrupt routine that is called after REGTRN has enabled the proper interrupt and the master station is ready to accept an initial word.

Subroutines SENDIW, REMSTR, SENDDN, and SRDATA use TRNTBL locations 1 through 4.

Location	Description
1	Index pointing to location in TRNTBL containing the data to be transferred
2	Transfer control flag (a) 1 = initial contact for transfer (b) 2 = during a transfer (c) 3 = after the last transfer index is reached (d) 4 = one word transfers only (3) 5 = transfer complete

Location	Description
3	Storage for any control word received from the master (initial reply or interrupt word)
4	Contains the op-code for the transfer in progress bits (8-15) and the ISUBID of the subroutine requesting the transfer bits (0-7)

The initial word is located by indexing into TRNTBL, using the index stored in TRNTBL(1). After the initial word is transmitted, TRNTBL(1) is then incremented to point at the first device number.

The flow diagram for send the initial word (SENDIW) is shown in Figure 22.

2.2.2.2.4 Send Device Number (SENDDN)

This routine is an interrupt-called routine that outputs the device number found by using the index at TRNTBL(1). After the device number is transmitted, the index is then incremented to point at the next data location.

The send device number (SENDDN) flow diagram is shown in Figure 23.

2.2.2.2.5 Send and Receive Data (SRDATA)

This routine is an interrupt-called routine that either outputs data or stores data as directed by the type of interrupt (which half of the DR11 is interrupted). Again TRNTBL(1) is used as the index for output or storing and is incremented to point at the next device number (if required). (See Figure 24.)

2.2.2.2.6 Read Master Station (REMSTR)

This routine is an interrupt-called routine used for inputing the initial replies or interrupt words from the master station. All of these responses are directed by the master station to the initial word interface unit.

There are seven functions performed by this routine.

a. Input word from master station.

b. Check the transfer control flag [TRNTBL(2)]. (If this flag is out of range, a STOP abort is performed, as interrupt troubles have occurred.)

c. Decide if word is an interrupt or initial reply.

d. If an interrupt, determine if it is a scheduled or non-scheduled interrupt.

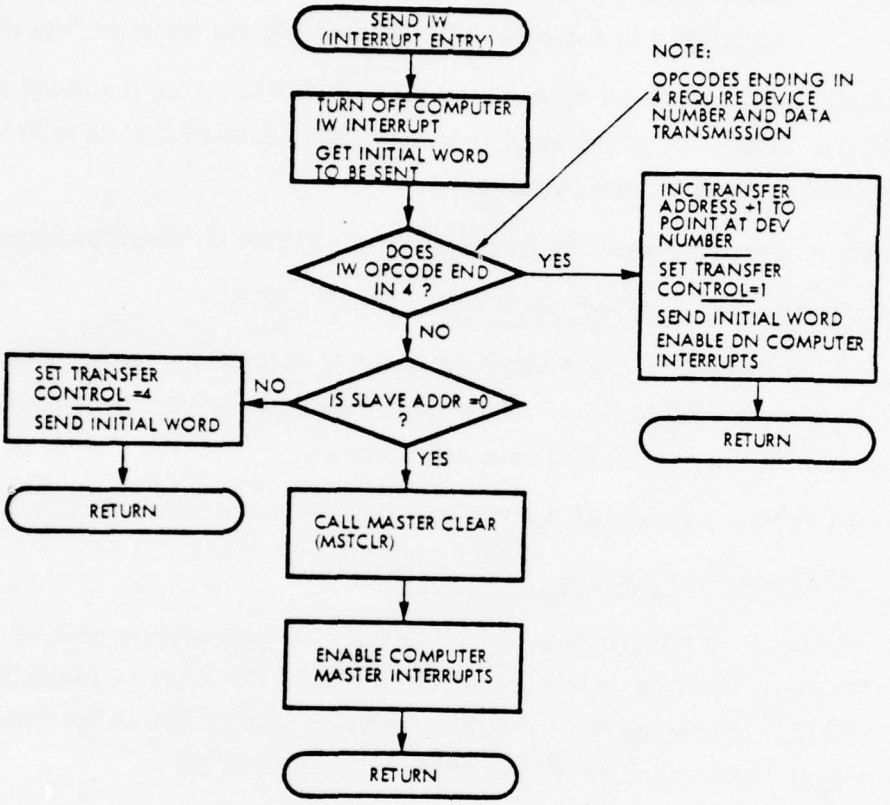


Figure 22. Send the Initial Word (SENDIW), Flow Diagram

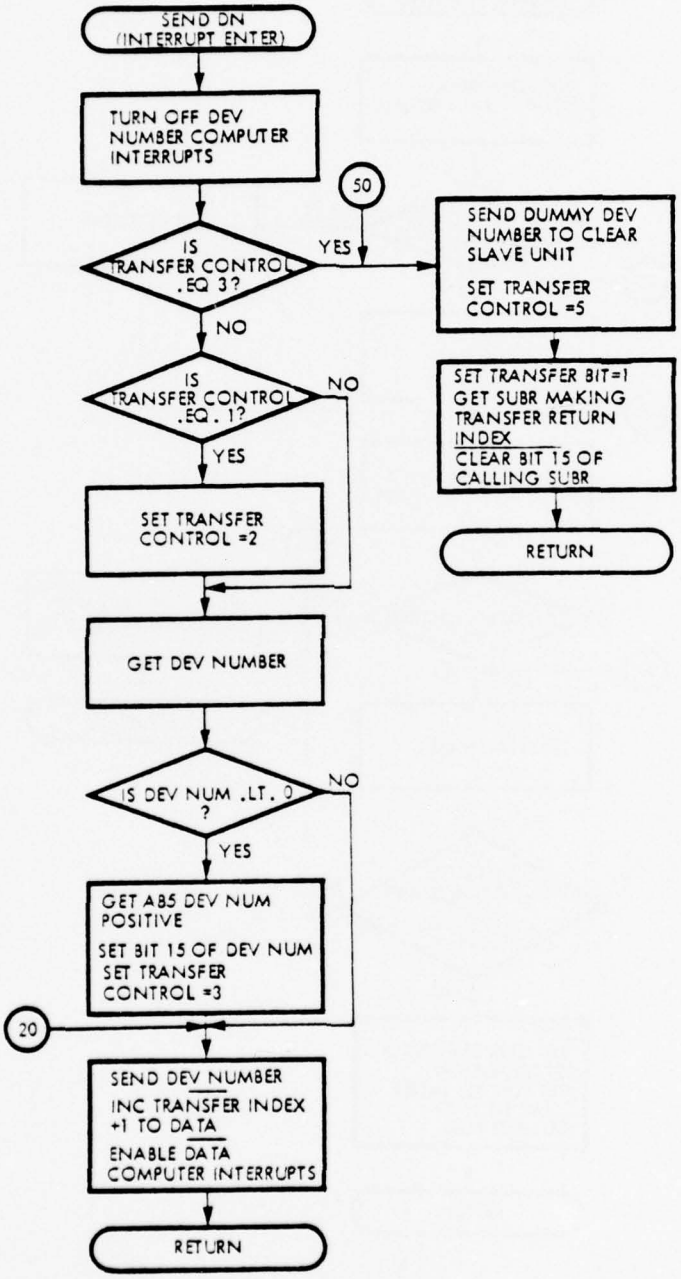


Figure 23. Send Device Number (SENDDN), Flow Diagram

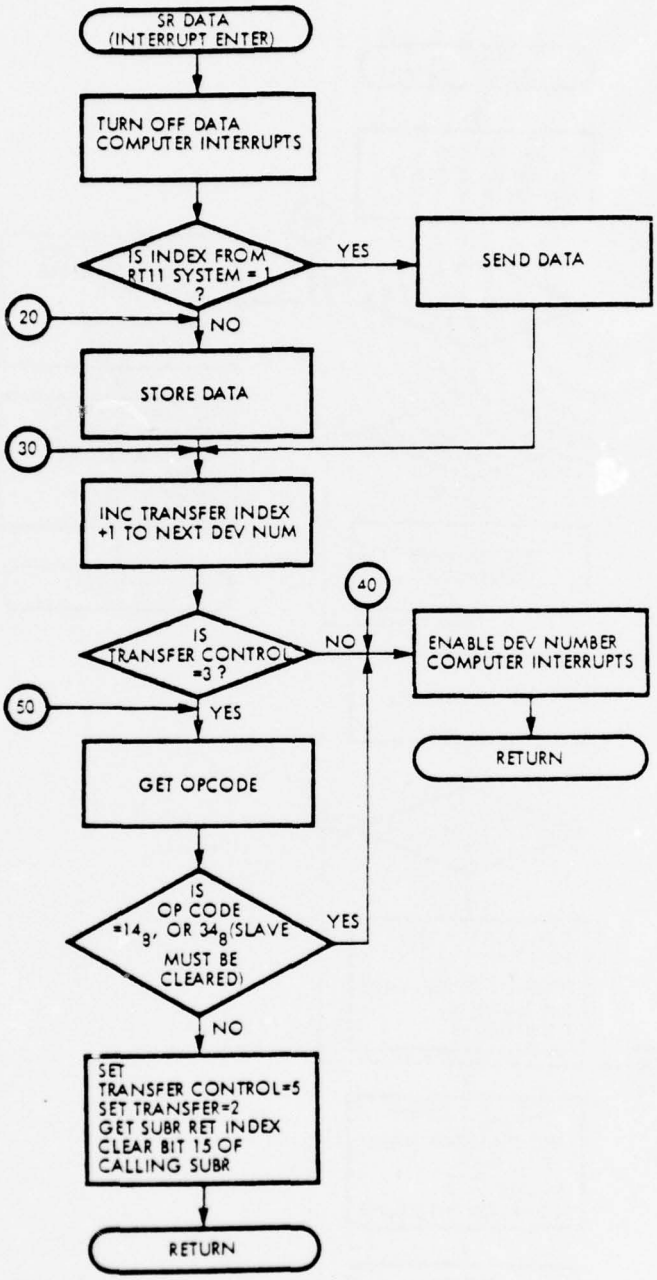


Figure 24. Send and Receive Data (SRDATA), Flow Diagram

e. If a scheduled interrupt:

(1) Obtain the ISUBID of the requesting routine.

(2) Store the interrupt word in the CYCTBL data entry for the routine.

(3) Clear bit 15 of the requesting routine CYCTBL cycle control entry.

(4) Clear the bookkeeping records.

(5) Return.

f. If a non-scheduled interrupt:

(1) Notify error report (ERRRPT).

(2) Clear bookkeeping records.

(3) Return.

g. If an initial reply:

(1) Determine if an error occurred; if so, increment error count in the requesting subroutine CYCTBL cycling control word.

(2) Determine if initial reply is to be ignored or returned to requesting subroutine, and take appropriate action.

(3) Clear bookkeeping records.

(4) Return.

The flow diagram for Read Master (REMSTR) is shown in Figure 25.

2.2.2.2.7 Direct Memory Access Interrupt (DMAINT)

This routine is an interrupt-called routine at the completion of a DMA transfer.

There are three major functions.

a. Check for error returns from the DR11B; if error return, increment error count for the requesting subroutine.

b. Clear bit 15 of the cycling control word for the requesting subroutine.

c. Return.

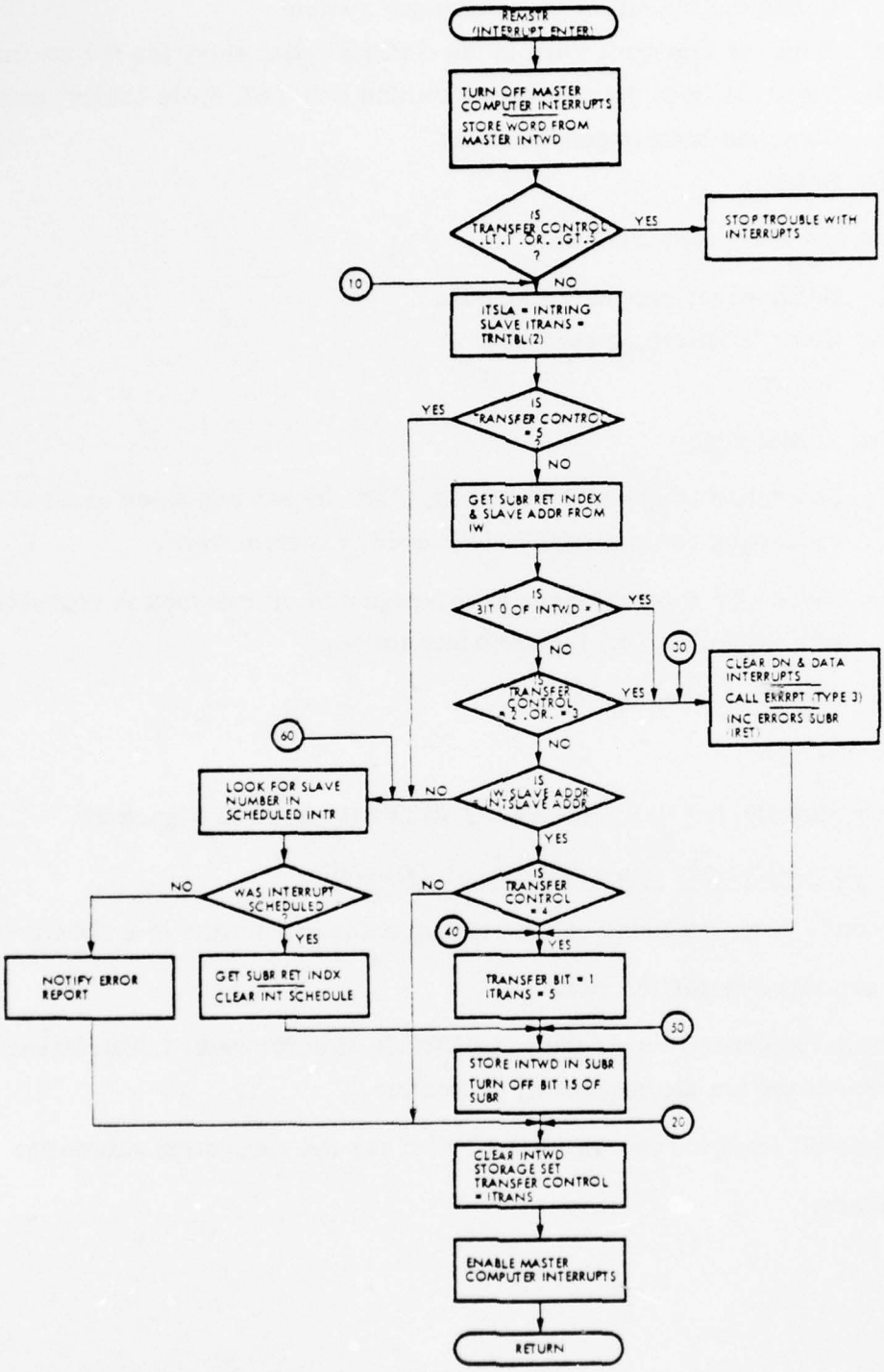


Figure 25. Read Master (REMSTR), Flow Diagram

2.2.2.3 Slave Station Control

The following subroutines are designed to perform the two basic control functions available in the slave units, namely, clearing of the local slave unit logic and requesting the master/slave units to perform a communications check for line driver/receiver errors.

2.2.2.3.1 Local Clear (LCLCLR)

There are four primary purposes of this subroutine.

a. Generate the op-code for a local clear.
b. Add the requested slave unit address.
c. Initiate the transfers.
d. Control the initial reply.

2.2.2.3.2 Manual Local Clear (MANCLR)

The purpose of this routine is to allow the operator access to LCLCLR through the system console keyboard.

2.2.2.3.3 Communications Check (COMMCK)

There are four primary purposes of this subroutine.

a. Generate the op-code for a communications check.
b. Add the requested slave unit address.
c. Initiate the transfer.
d. Control the initial reply.

2.2.2.3.4 Manual Communications Check (MANCMK)

The purpose of this routine is to allow the operator access to COMMCK through the system console keyboard.

2.2.2.4 System Initialization

The following group of subroutines are used to allow the operator to determine the system parameter desired for the operation or experiment. The operator may select entirely new parameters or old parameters (recorded on disk), or may modify the old parameters.

2.2.2.4.1 New Start (NSTART)

There are four functions of this subroutine.

a. Interface with the operator through the Tektronics graphic display and the keyboard.

b. Initialize all system tables with either zero's or recorded data, depending upon operator.

c. Call the proper programs:

(1) MLKCON (see Paragraph 2.2.2.1.2).

(2) Field function control (FFUCON).

d. Record the selected system parameters for future use.

2.2.2.4.2 Field Function Control (FFUCON)

The purpose of this routine is to interface with the operator, recall the field function display file, and display on the GT44 graphic display a picture of the field function (see Figure 26). The operator selects the field logic switching desired (display graphically changes the display to reflect the position of the switches). Upon completion by the operator, the routine returns to NSTART for recording the desired field relay logic configuration in a single word (FLDCWD).

The configuration selected during this routine is for each of the 384-element field logic units; hence, calibration 1 is locked out so that it may not be selected. The reason for this is that the calibration 1 is on a per-element basis only.

2.2.2.4.3 Display Control (DPYCON)

This routine is used by FFUCON to actually display the picture of the field logic functional diagram and to control the subpictures of the switch positions. In addition, this routine supports the element function control (EFUCON) subroutine (Paragraph 2.2.2.6.1).

2.2.2.4.4 Roll Call (ROLCAL)

This routine indexes through all slave addresses, calls for a communication check (COMMCK), and reports the results to the operator on the Tektronix graphics terminal.

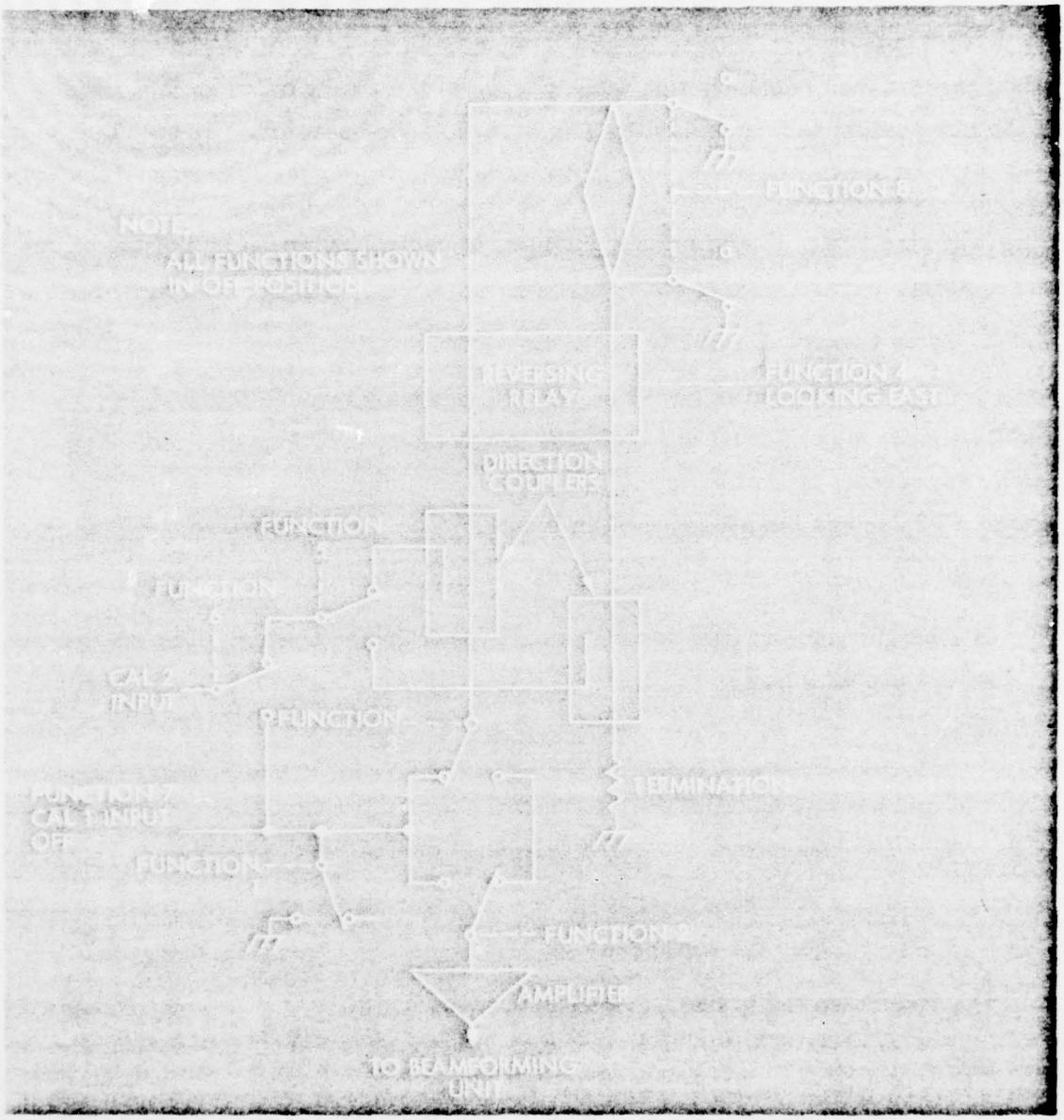


Figure 26. Graphic Display of Field Function Control (FFUCON)

2.2.2.4.5 Initialize Field (INTFLD)

This routine uses the field logic configuration stored in FLDCWD to generate an initial word and a table of device numbers and data words for the 96 locations in each of four slaves.

The program then transfers this table to each of the four slaves, in turn, and reads the information back and displays any errors to the operator.

2.2.2.5 System Control

The following group of subroutines are used to control the operation of the beam-forming system.

2.2.2.5.1 Cycle Control (CYCCTL)

This routine is designed to interface with the operator through the system console keyboard, thus allowing the operator to select the functions desired. The selection, at present, is limited to the following nine items.

(Note: "#" denotes the presence of an integer value that is required.)

a. AB#

Is a maintenance routine for checking Type A driver boards. The integer values are as follows:

Integer	Description
1	No readback
2	Readback and error displaced on Tektronix graphic terminal and LA36 bell
3	Stop the routine.

The routine called is NSTEP with an op-code of 16.

b. BB#

Is a maintenance routine for checking Type B driver boards. The integer values are the same as for AB#. The routine called is NSTEP with an op-code of 8.

c. CB#

Is a maintenance routine for checking Type C driver boards. The integer values are the same as for AB#. The routine called is NSTEP with an op-code of 10.

d. CK#[###]

Calls the slave unit command routine MANCMK. The value of the first integer is as follows.

Integer	Description
1	A single communications check is made with the indicated slave unit.
2	Continuous communications checks are made with the indicated slave unit.
3	Continuous communications checks are stopped.

The three following integers are the desired slave unit. (Only one, two, or all three values are required, depending on the slave address.)

e. ET#

Is a maintenance routine for checking the field logic operation. The routine called is element function control (EFUCON). The integer value is immaterial, but a single number must be present.

f. FS#[##]

Is a maintenance routine similar to BB#, except the integer values in the brackets must be the column number desired. The first integer value has the same meaning as in AB#. The routine called is NSTEP with an op-code of 8 plus the column number. (Only one digit is required for values below 10.)

g. LC#[###]

Calls the slave unit command routine MANCLR. The values of the integers are the same as for CK#[###].

h. STOP

Returns the program to the operating system, where the operating system then executes the abort table supplied to turn off all interrupts.

i. TS#

Calls the synthesizer tuning (SYNTUN) subroutine. The integer value is immaterial at present, since only one synthesizer was interfaced; however, a number must be present.

2.2.2.5.2 Synthesizer Tuning (SYNTUN)

There are four functions of this routine.

a. Call GFREQ for the frequency required in the BCD, packed according to Paragraph 2.1.5.3.3.

b. Transmit the data to the proper synthesizer.

c. Read back the data and compare for errors; if any, report to the operator and the calling subroutine.

d. Return.

The flow diagram for synthesizer tuning (SYNTUN) is shown in Figure 27.

2.2.2.5.3 Get Synthesizer Frequency (GFREQ)

The function of this routine is to interface with the operator to obtain the desired frequency, to tune the synthesizer, and to pack the resulting frequency in BCD form in the output table.

The get synthesizer frequency (GFREQ) flow diagram is shown in Figure 28.

2.2.2.6 Diagnostics and Maintenance

The following group of subroutines are for diagnostics and maintenance of the system and individual elements of the system.

2.2.2.6.1 Element Function Control (EFUCON)

This routine is similar to the field function control (FFUCON) routine, except all field logic changes are transmitted for each change, as requested, and the calibration 1 system can be used. Thus, the operator can change the logic of one element and observe the results as he changes configurations.

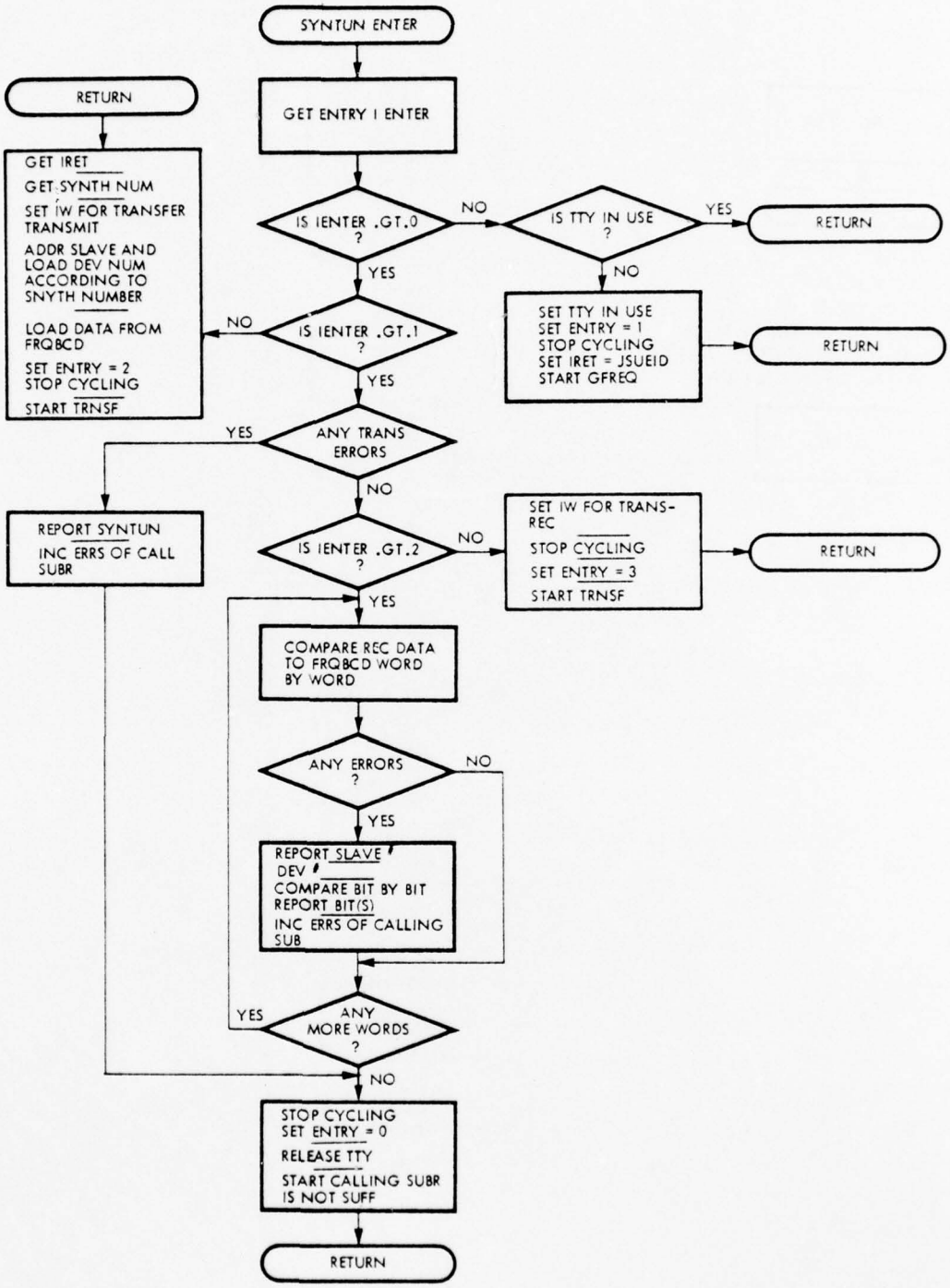
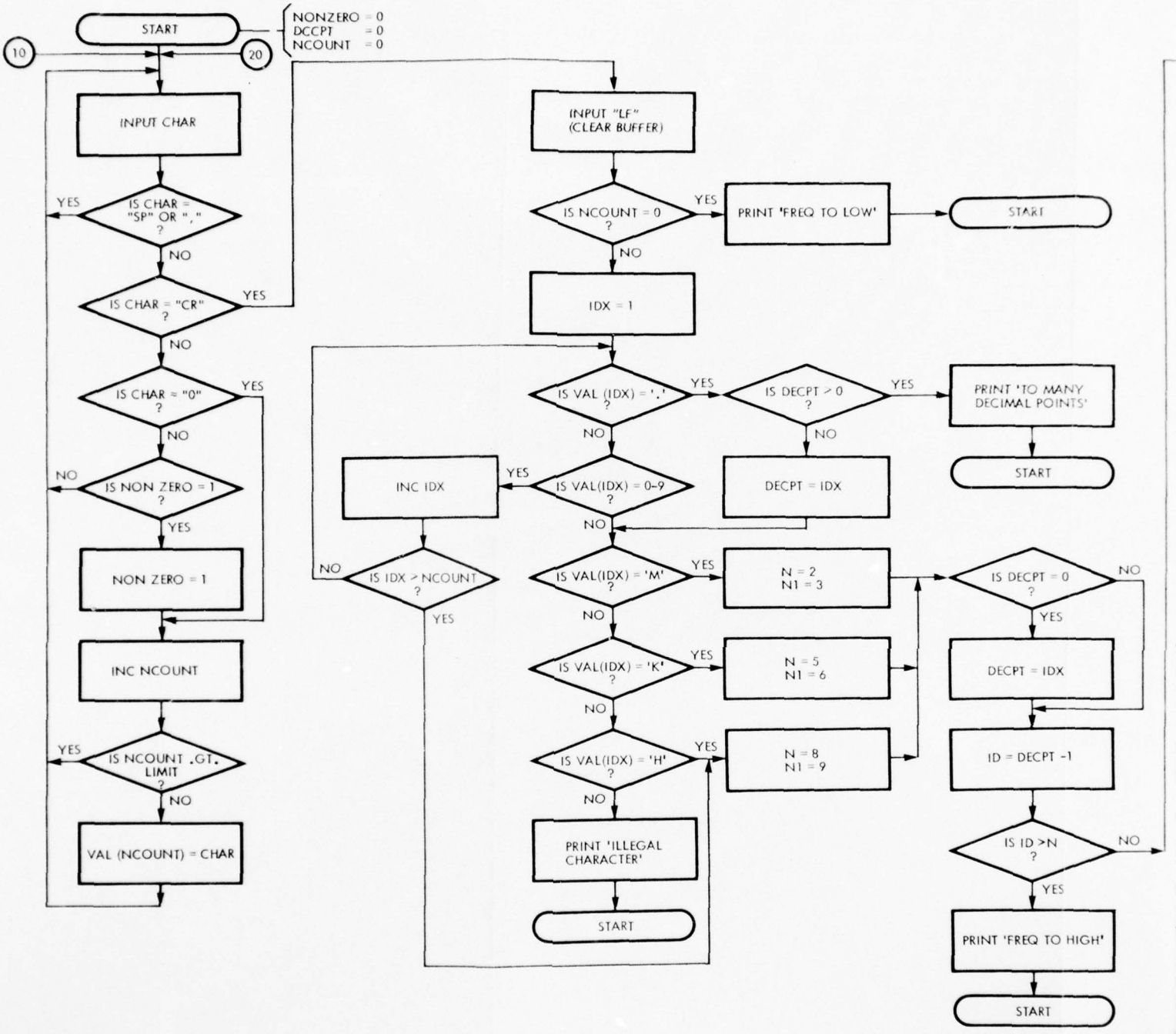


Figure 27. Synthesizer Tuning (SYNTUN), Flow Diagram


START
NONZERO = 0
DCCPT = 0
NCOUNT = 0
10
20
INPUT CHAR
IS CHAR = "SP" OR "," ?
YES
NO
IS CHAR = "CR" ?
YES
NO
IS CHAR = "0" ?
YES
NO
IS NON ZERO = 1 ?
NO
YES
NON ZERO = 1
INC NCOUNT
IS NCOUNT .GT. LIMIT ?
YES
NO
VAL (NCOUNT) = CHAR
INPUT "LF" (CLEAR BUFFER)
IS NCOUNT = 0 ?
YES
PRINT 'FREQ TO LOW'
START
NO
IDX = 1
IS VAL (IDX) = '.' ?
YES
IS DECPT > 0 ?
YES
PRINT 'TO MANY DECIMAL POINTS'
START
NO
DECPT = IDX
NO
IS VAL(IDX) = 0-9 ?
YES
INC IDX
NO
IS IDX > NCOUNT ?
YES
NO
IS VAL(IDX) = 'M' ?
YES
N = 2
N1 = 3
NO
IS VAL(IDX) = 'K' ?
YES
N = 5
N1 = 6
NO
IS VAL(IDX) = 'H' ?
YES
N = 8
N1 = 9
NO
PRINT 'ILLEGAL CHARACTER'
START
IS DECPT = 0 ?
NO
YES
DECPT = IDX
ID = DECPT -1
IS ID >N ?
NO
YES
PRINT 'FREQ TO HIGH'
START

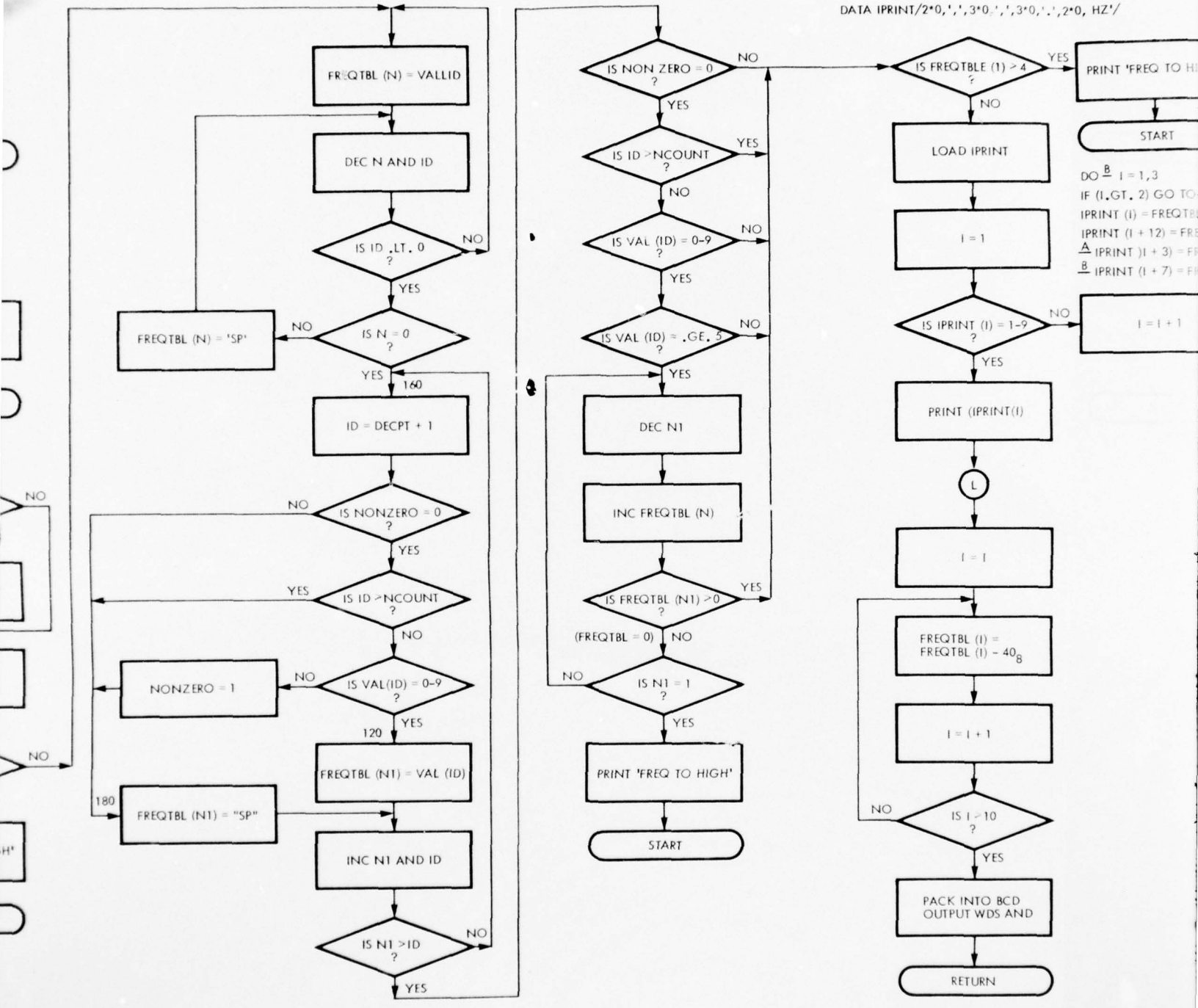


Figure 28. Get Synthesizer F
(GFREQ), Flow D

2

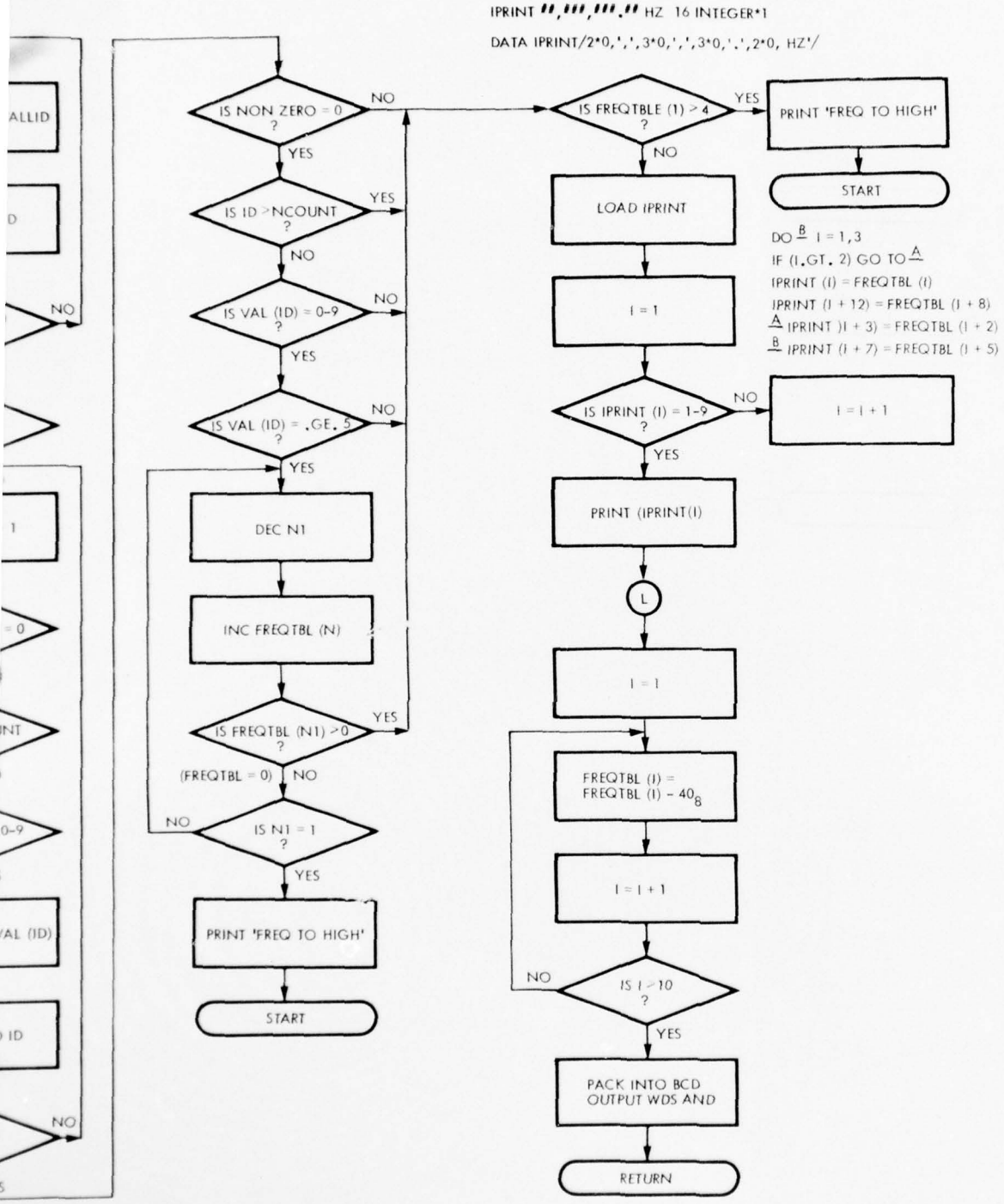


Figure 28. Get Synthesizer Frequency (GFREQ), Flow Diagram

This program uses DPYCON for control of the actual display (which will now include the column and row number being controlled) and exits back to the main program through initialize element (INTELM).

2.2.2.6.2 Initialize Element (INTELM)

This routine uses the field configuration word FLWCWD to restore the element used by EFUCON back to the same configuration as the other 383 elements. Thus, it is impossible to leave one element configured in a different way from the others.

2.2.2.6.3 Relay Driver Board Tests (NSTEP)

This is a maintenance routine to perform the tests as outlined in Paragraph 2.2.2.5.1 (AB#, BB#, CB#, and FS[##]. This routine exits through the initialize element (INTELM).

The flow diagram for NSTEP is shown in Figure 29.

2.2.2.6.4 Slave Unit Bit Test (BITTST)

This is a diagnostic routine to check each bit in every slave unit (a) by using two bit patterns (odd/even) which are transmitted individually to each device number while all other device numbers have all bits set, and (b) then again with all bits off.

All errors found are reported, giving the pattern, bit status, bit in error, slave address, device address, and board position number.

2.2.2.7 Support Routines

The following group of five routines are used to support other routines by doing a single function.

2.2.2.7.1 Device Number (Integer) to BCD (DEVBCD)

This routine is to provide a BCD word in place of an integer value. In addition, the function will return the BCD word with or without bit 15 set (not negated). The formal parameters are IDEV (the integer value) and ITYPE for a value of zero, bit 15 is left clear; for a value of 'one,' bit 15 is set on RETURN.

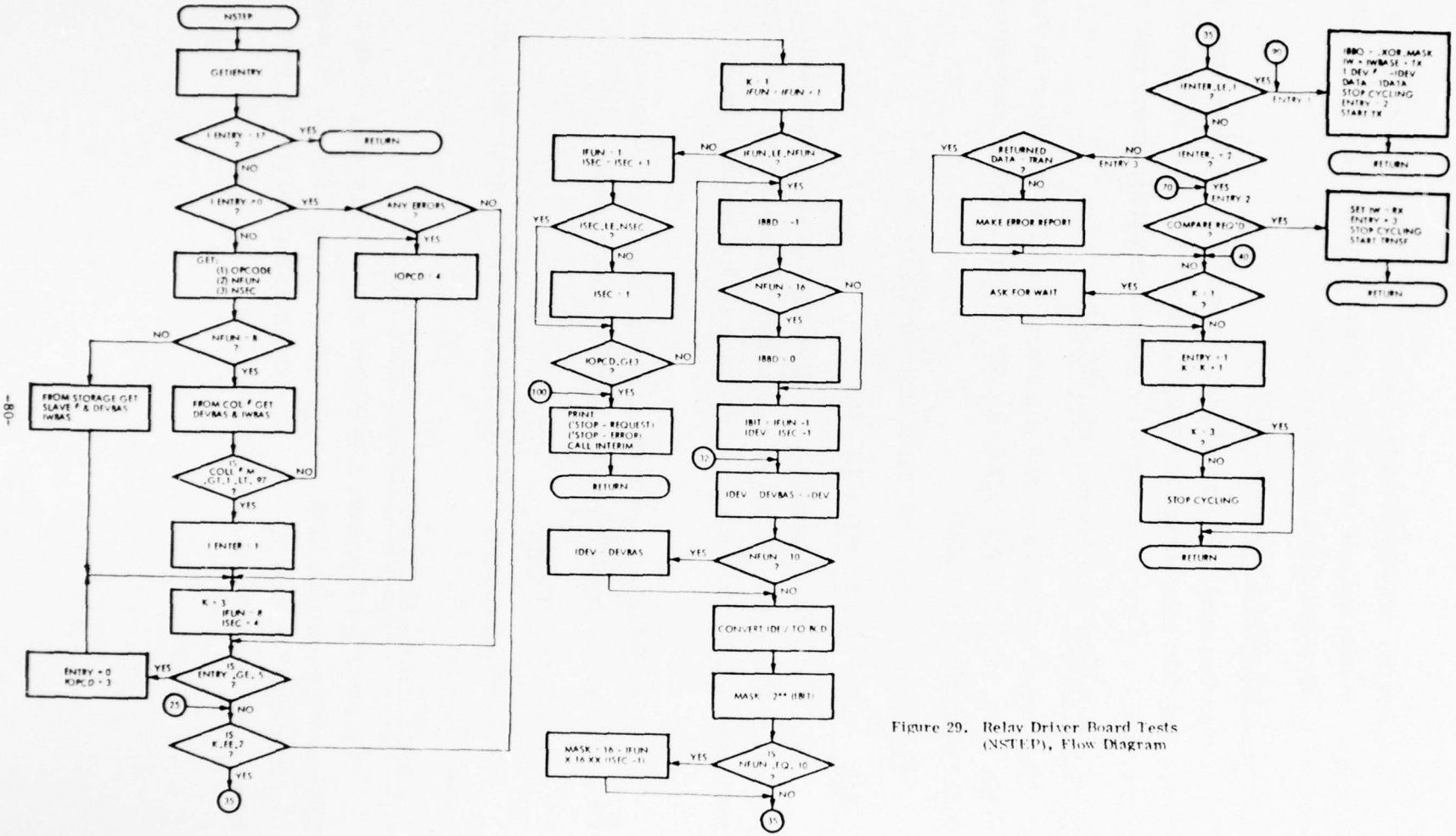


Figure 29. Relay Driver Board Tests (NSTEP), Flow Diagram

2.2.2.7.2 Field Logic Bit Configuration Decoding (FLDDEC)

This routine is required since the physical configuration (nearly 3100 cables between the slave units and the field logic units) dictated scrambling the storage location of field function bits in the Type B driver, instead of in a logical sequence.

The bit pattern repeats every 32 bits or at each column station in the field, which is four device numbers or one Type B driver. Thus, the field configuration word FLDCWD, which is stored logically, must be decoded in groups of four to locate the proper position for the controlling bit.

The formal input parameters are NROW and NFUN:

a. NROW Row number of the element (1-4)
b. NFUN Function number (1-8)

The routine returns are IDEV and NBIT.

a. IDEV The device number containing the desired function (0-3)
b. NBIT The bit number (1-8) that controls the function.

The routine uses a table look-up. The table is stored on RK0 disk in file STRVAL.DAT. If the proper block number from STRAV.DAT (0-99) is not core resident, the routine will read block zero from the disk into core before proceeding.

2.2.2.7.3 Restart Timing (TIMER1)

This subroutine is used to take advantage of the RT11-V02B operating system SYSLIB function timer. When a subroutine is required to operate not on an event basis but on a time base (e.g., every three seconds), a call is made to timer, with the specifications for HRS, MIN, SEC, TICKS, identification number, and the name of the subroutine. TICKS is 16-2/3 ms (line frequency clock interval).

The call is made using TIMER1 as the selected subroutine and the identification number equal to the ISUBID desired to be cycled. At the expiration of the specified time interval, TIMER1 is run by the RT11-V02B operating system, which then clears cycling bit 15 of the desired subroutine.

AD-A038 135 GTE SYLVANIA INC MOUNTAIN VIEW CALIF ELECTRONIC SYST--ETC F/G 17/9
ADVANCED ARRAY DESIGN, TEST AND EVALUATION.(U)
JAN 77 J E RICE

UNCLASSIFIED F30602-74-C-0252
RADC-TR-77-25 NL

2 OF 2
AD
A038135

END

DATE
FILMED
5-77

2.2.2.7.4 Error Report Decoding (ERRRPT)

This subroutine is used to decode and print the type and the report error, if any, from a slave unit reply (either initial reply or interrupt reply). This decoding is done in accordance with the error reporting table shown in Paragraph 2.1.4.5.

The error report decoding (ERRRPT) flow diagram is shown in Figure 30.

2.2.2.7.5 Beamformer (Broadside) Logic Bit Configuration Decoding (HSEOUT)

In the same fashion that physical characteristics dictated bit usage in the field, the broadside trimming units were similarly scrambled.

The subroutine uses the identical logic as field logic decode (FLDDEC), except the pattern repeats every 16 bits, thus halving the table size.

The one exception this program must accommodate is that two slave units (3 and 4) were physically wired in a different manner from the others. Therefore, in the range of bits (0-7), a separate table is maintained for units 3 and 4.

2.3 ANCILLARY PROGRAMS

This group of six programs, while not part of program DEXTER, was required to initiate the recorded (disk) files used by DEXTER.

2.3.1 STARTS

This program is used to place data constants into disk storage for use by DEXTER (e.g., the field logic bit decoding tables). In addition, it provides the entry points for these tables and length for expanding the stored data with new constants as they become available.

2.3.2 Master Station Linkage Display (MLKDPY)

The program was used to generate the GT44 picture display of the master station units. The program then recorded the picture on disk RK0 for future use as file MLKDPY.DPY.

2.3.3 Field Function Display (FFUDPY)

This program functioned exactly as MLKDPY, except the display picture was of the field logic functional diagram and the recorded file is FFUDPY.DPY.

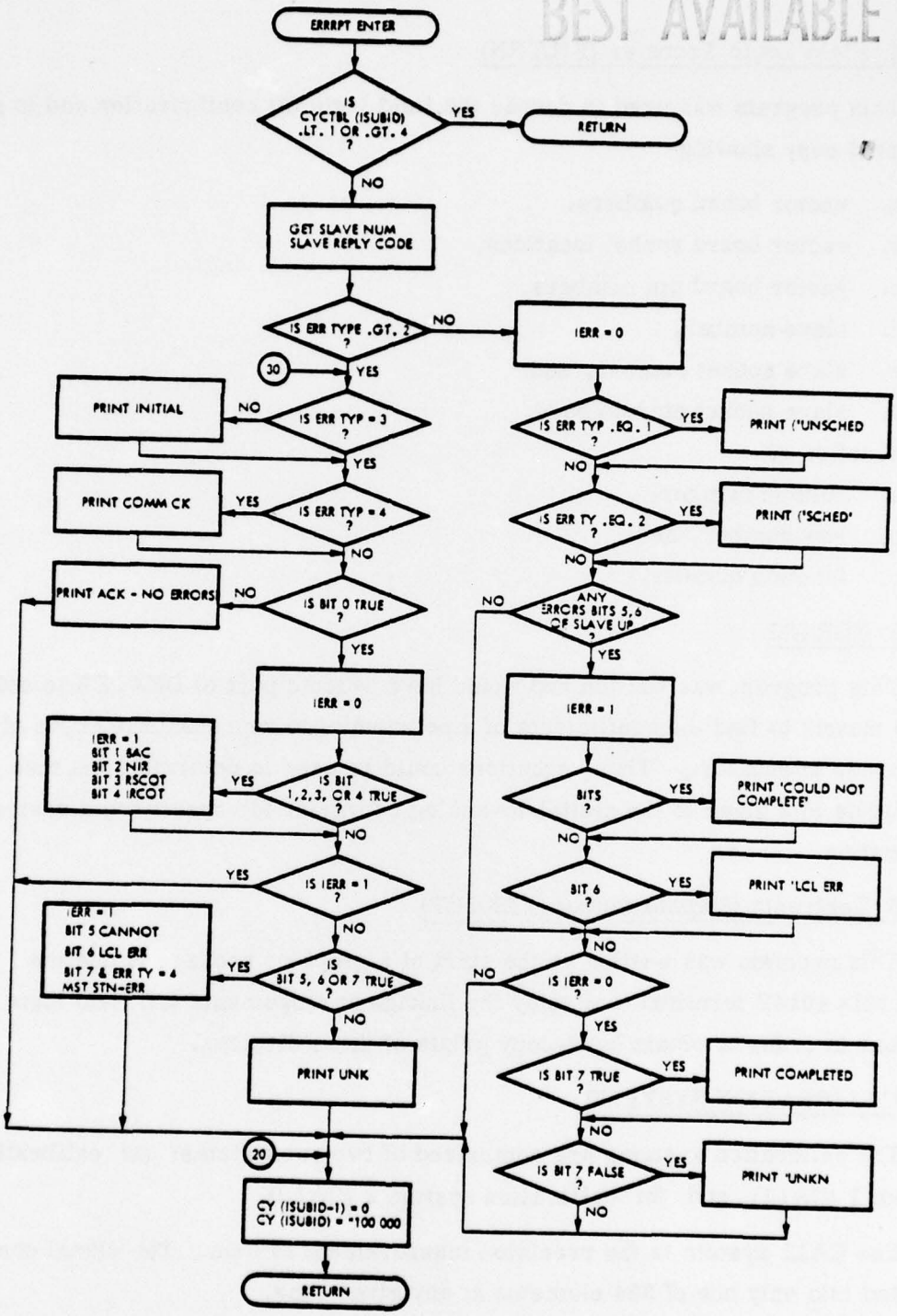


Figure 30. Error Report Decoding (ERRRPT), Flow Diagram

2.3.4 Field Logic Transfer (FILTRN)

This program was used to decode the field logic bit configuration and to provide a printed copy showing

a. vector board numbers,
b. vector board socket locations,
c. vector board pin numbers,
d. slave number,
e. slave socket number, and
f. slave socket pin numbers,

as a function of,

a. column number,
b. row number, and
c. function number.

2.3.5 NORTST

This program was written and would have become part of DEXTER to solve a 9x9 matrix to find the coefficients of a polynomial to represent the curve of the electronic attenuators. Thus, equations could be used to determine the nine magnitude bits given to the digital-to-analog converter for specifying a desired attenuation.

2.3.6 Tektronix Graphic Output (TEKOUT)

This problem was written as the start of a graphics package to use the Tektronix 4014E terminal to display the linkage arrangements and field logic configurations in order to obtain hard-copy prints of these displays.

2.4 CALIBRATION SYSTEMS

The calibration systems are comprised of two subsystems: (a) calibration system 1 (CAL1), and (b) calibration system 2 (CAL2).

The CAL1 system is the precision measurement system. The signal can be injected into only one of 384 elements at any given time.

The limit of the beamformer performance will be set by the accuracy of the CAL1 system at the lower frequencies and by the trim units at the higher frequencies.

The estimated performance of the beamformer is shown in Figure 31, and the estimated performance of the array (at 17.97 MHz using the measured antenna pattern) is shown in Figure 32. These two figures indicated the expected tolerance limits set by the trim units and CAL1 system.

The CAL2 system is used to inject a signal simultaneously into all 384 elements, thus providing a system that can duplicate a boresight signal.

2.4.1 Calibration System CAL2

The CAL2 system is composed of three binary trees of five levels using two-way power splitters. The three binary trees follow a single three-way splitter. This produces 96 simultaneous outputs. These 96 outputs then individually feed 96 series strings consisting of directional couplers. The 384 outputs of the directional couplers are the CAL2 input to the field logic units located at the elements. The block diagram of the CAL2 system is shown in Figure 33.

2.4.2 Calibration System CAL1

The CAL1 system is composed of two major subsystems.

a. The CAL1 distribution system
b. The calibration measurement system

2.4.2.1 CAL1 Distribution System

The CAL1 distribution system consists of two subsystems.

a. A binary switch unit located in the beamformer racks

b. A series switching system located in the field as part of the field logic unit

The binary switch unit directs the input signal to one of 16 cables, which then feeds out of the building to a group of six columns. The six columns are fed through switches in series. The selection of any one column then removes the CAL1 signal from the series line and feeds four switches, in series, to each row within the column. This column selection is done in parallel when a given column/row element selection is made. Thus, the CAL1 distribution has 384 outputs; however, only one of the 384 can be selected at any given time. The block diagram of the CAL1 distribution system is shown in Figure 34.

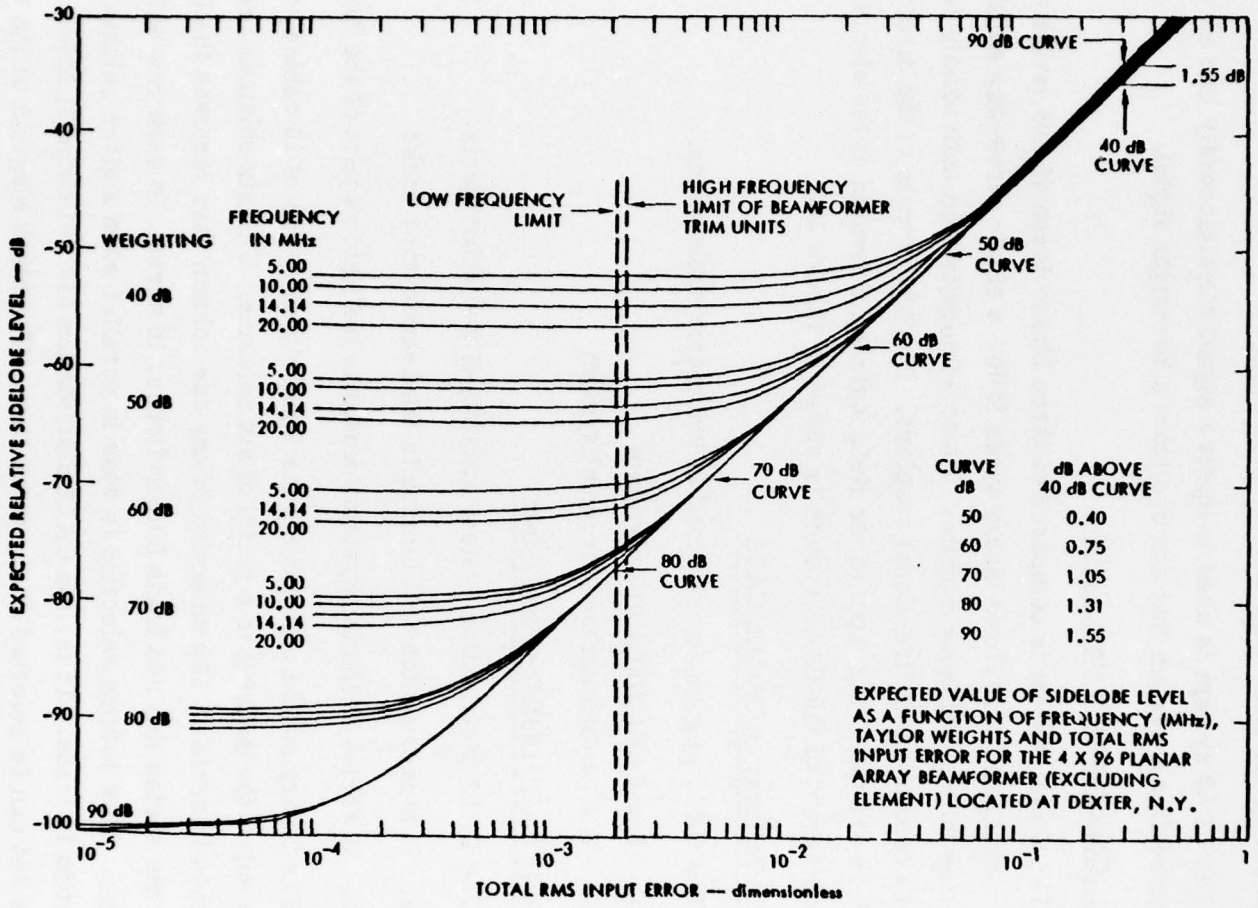


Figure 31. Estimated Performance of the Beamformer

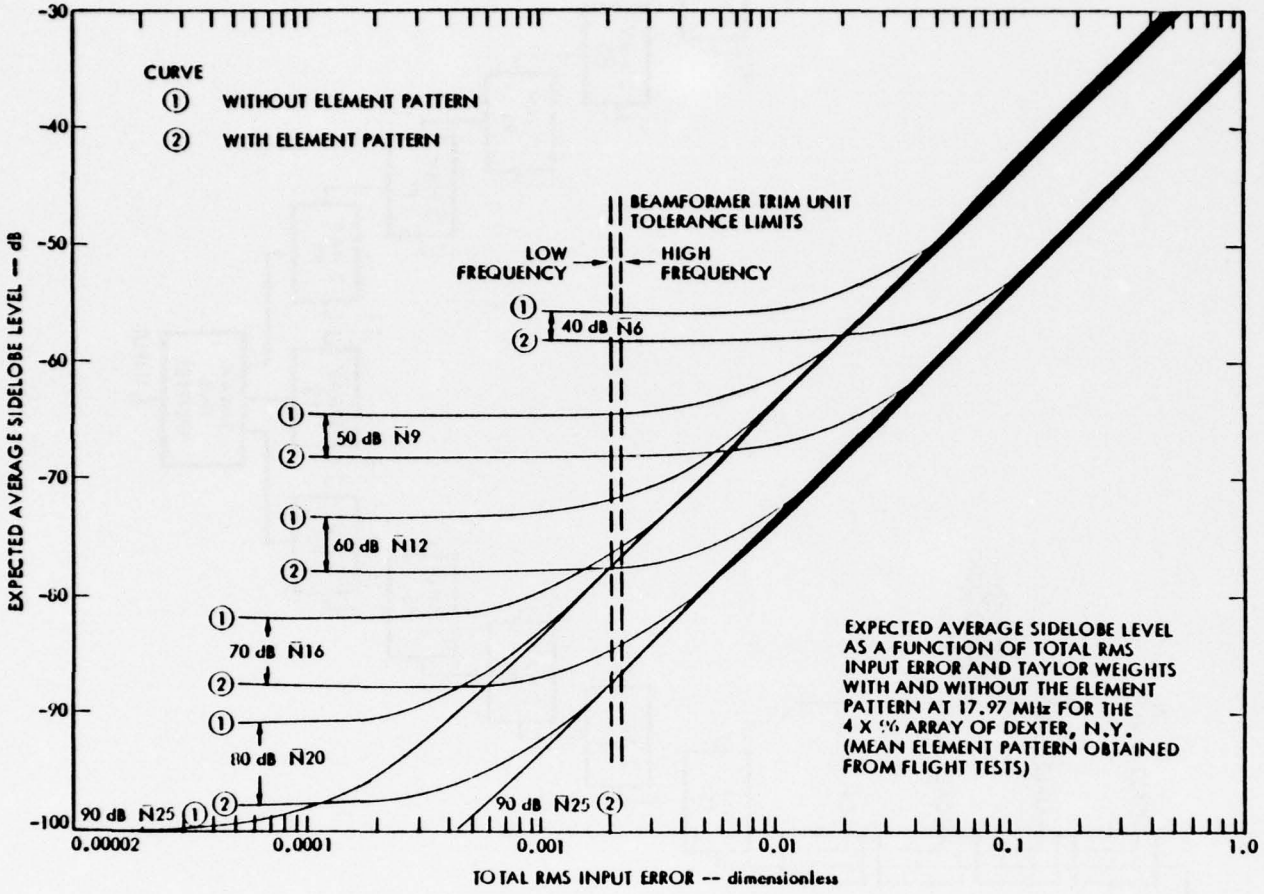


Figure 32. Estimated Performance of the Array at 17.97 MHz Using the Measured Antenna Pattern

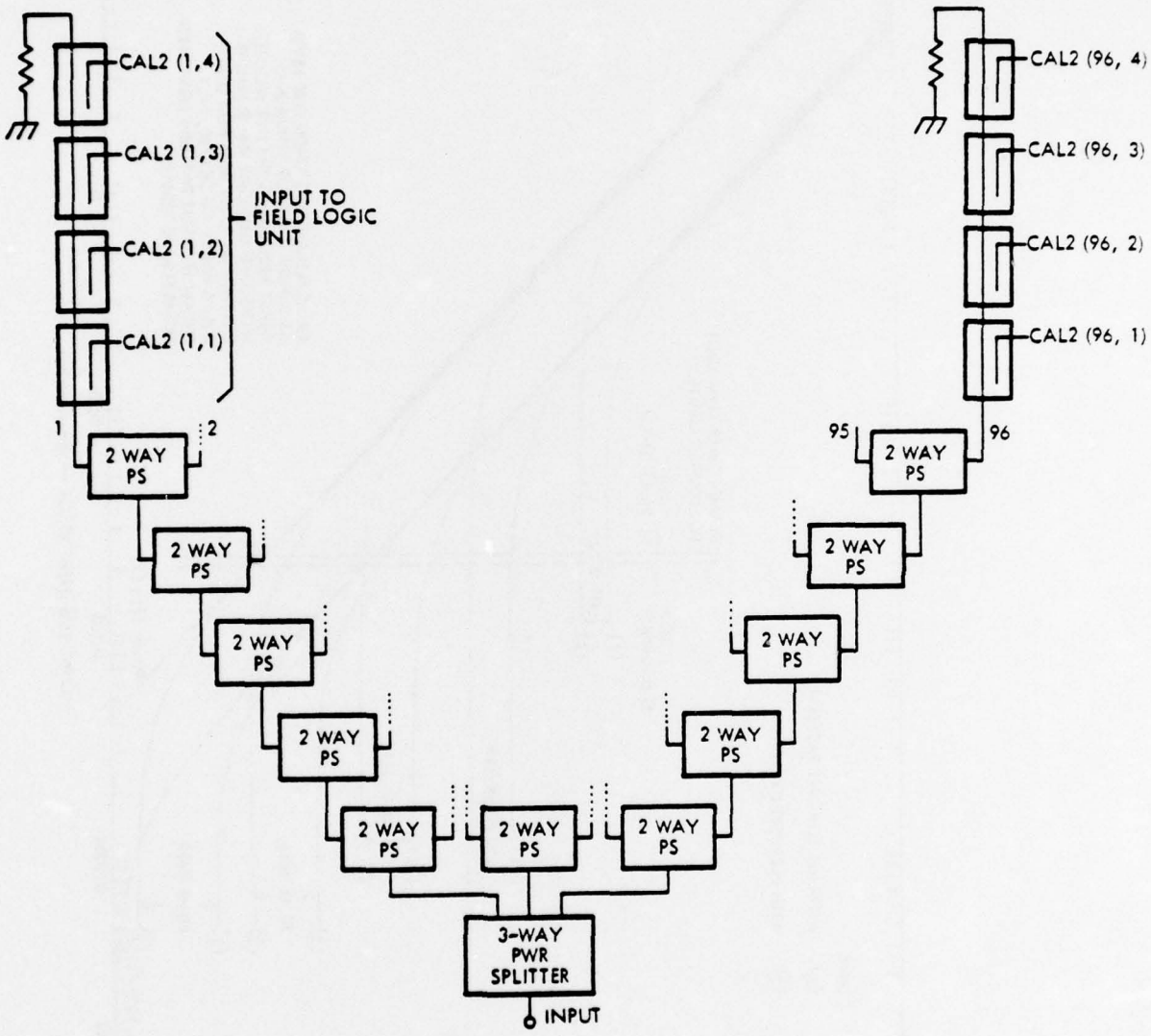


Figure 33. Calibration System Number Two (CAL2), Block Diagram

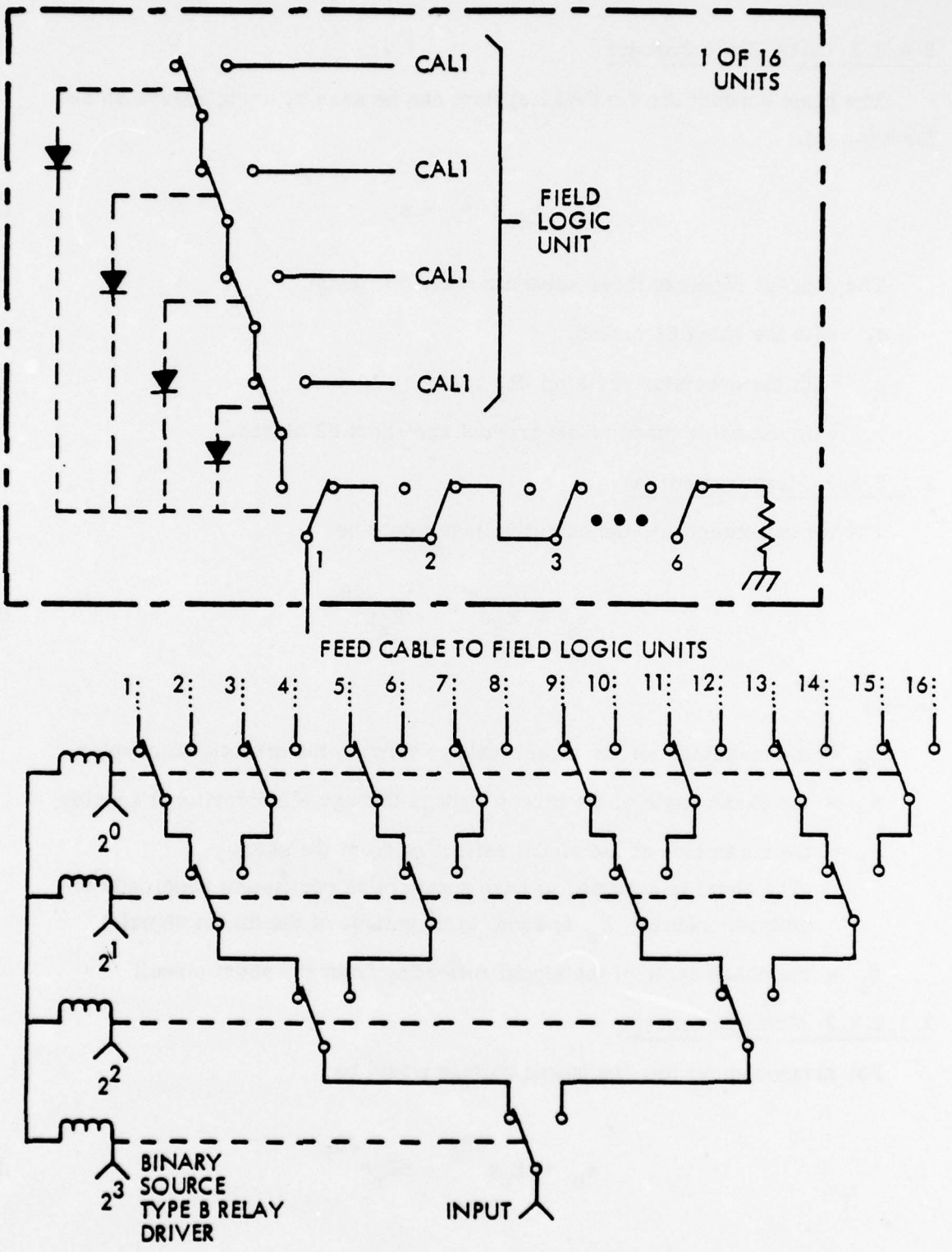


Figure 34. Calibration System Number One (CAL1) Distribution System, Circuit Diagram

2.4.2.2 CAL1 Basic Concept

The basic concept for the CAL1 system can be seen by using Figure 35 and Equation (1).

$$e_o = e_d + e_r \tag{1}$$

The concept requires three separate measurements:

a. with the short S1 closed,

b. with the precision resistor R_L, and

c. with the cable (under test) present and short S2 closed.

2.4.2.2.1 Measurement (a)

For measurement (a), the output voltage would be

$$e_a = E_d e^{J\theta_d} + E_s e^{J\theta_s} \tag{2}$$

where:

E_d = the magnitude of the direct leakage through the directional coupler

θ_d = the phase angle of the direct leakage through the directional coupler

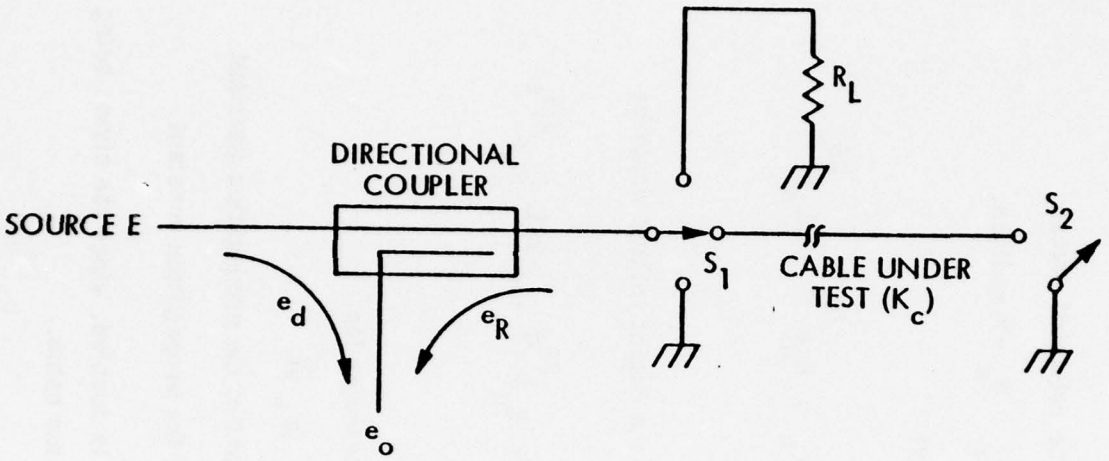
E_s = the magnitude of the signal reflecting from the short
(The short is assumed to have a reflection coefficient magnitude of unity; therefore, E_s is equal to magntiude of the direct signal.)

θ_s = the phase angle of the signal reflecting from the short circuit

2.4.2.2.2 Measurement (b)

For measurement (b), the output voltage would be

$$e_b = E_d e^{J\theta_d} + \rho E_s e^{J\theta_r} \tag{3}$$



e_d = DIRECT LEAKAGE SIGNAL (NOMINALLY 40–45 dB (BELOW THE REFLECTED SIGNAL)
e_o = DIRECTION COUPLER OUTPUT SIGNAL
e_R = REFLECTED SIGNAL FROM THE VARIOUS INTERFACING UNITS

Figure 35. CALI Basic Concept

where:

$/\rho/$ = the magnitude of the reflection coefficient from the precision (non-inductive 0.01%) resistor

$$/\rho/ = \left| \frac{R_L - Z_o}{R_L + Z_o} \right| \tag{4}$$

θ_r = the phase angle of the reflected signal

R_L = chosen such that $\theta_r = \theta_s + \pi$ radians.

Therefore, Equation (3) becomes

$$e_b = E_d e^{J\theta_d} - \rho E_s e^{J\theta_s} \tag{5}$$

2.4.2.2.3 Measurement (c)

For the measurement (c), the output voltage would be

$$e_c = E_d e^{J\theta_d} + e^{-2\gamma L} E_s e^{J(\theta_s)} \tag{6}$$

where:

L = the length of the cable under test

γ = propagation constant $(\alpha - j\beta)$

α = the attenuation constant of the propagation constant

β = the phase constant of the propagation constant

Note: The actual cable length is doubled, since the signal being measured travels down and back on the same cable.

Combining Equations (2) and (3) yields

$$E_s e^{J\theta_s} = \frac{e_1 - e_2}{(1 + \rho)} \tag{7}$$

Thus, the values

$$E_s = \left| \frac{e_a - e_b}{(1 + \rho)} \right| , \quad \theta_s = \arg \left\{ \frac{e_a - e_b}{(1 + \rho)} \right\}$$

and

$$E_d e^{J\theta_d} = e_a - E_s e^{J\theta_s}$$

are found.

Substituting these values in Equation (6) yields

$$e^{-2\gamma L} = \frac{(e_c - e_a)(1 + \rho)}{(e_a - e_b)} + 1 \tag{8}$$

Thus,

$$\alpha L = \frac{1}{2} \ell n \left| \frac{(e_c - e_a)(1 + \rho)}{(e_a - e_b)} + 1 \right| \tag{9}$$

and

$$\beta L = \frac{1}{2} \arg \left\{ \frac{(e_c - e_a)(1 + \rho)}{(e_a - e_b)} + 1 \right\} \tag{10}$$

Equations (9) and (10) do not allow the separation of αL and βL to obtain the three unknowns (α, β, L); they do give the value of the quantities actually desired. These are the loss in the cable and the electrical length in radians or degrees, so no further separation is required.

A linear error analysis was made on these equations to determine the relative accuracy with which the values e_a, e_b, e_c must be measured to obtain the design goal. To make RMS measurements of the cable constants of 0.03 dB and 0.1 degree) (assuming the measurement errors are random), the measurements must be made with the following tolerances:

a. ±0.0025 dB (amplitude)

b. ±0.005 degree (phase).

2.4.2.3 CAL1 Measurement System

To make measurements of the precision required, the CAL1 measurement system was designed and built (but not integrated or tested). The system was divided into two major sections.

a. CAL1 RF subsystem

b. Measurement subsystem

2.4.2.3.1 CAL1 RF Subsystem

The CAL1 RF subsystem is shown in Figure 36. The matching networks were provided to obtain high reverse isolation (greater than 90 dB achieved on the breadboard).

The frequency was dropped to 100 Hz to allow the precision attenuator to be constructed using 0.01% resistors in bridged-tee sections. The attenuator sections with 0.01% resistors provided an RMS error in attenuation of 0.0019 dB, well within the attenuation limits required.

2.4.2.3.2 CAL1 Measuring Subsystem

The CAL1 measuring subsystem is shown in Figure 37. The two sections, (a) amplitude and (b) phase, convert the analog signal to digital values. The analog section utilizes feedback to the precision attenuator to provide a signal of an amplitude within a narrow range about a given reference for the measurements. This narrow range of amplitude provides two advantages.

a. Use of the 12-bit analog-to-digital converter (ADC) as a high precision amplitude measurement device.

b. Prevention of phase errors in the phase equipment due to varying signal levels.

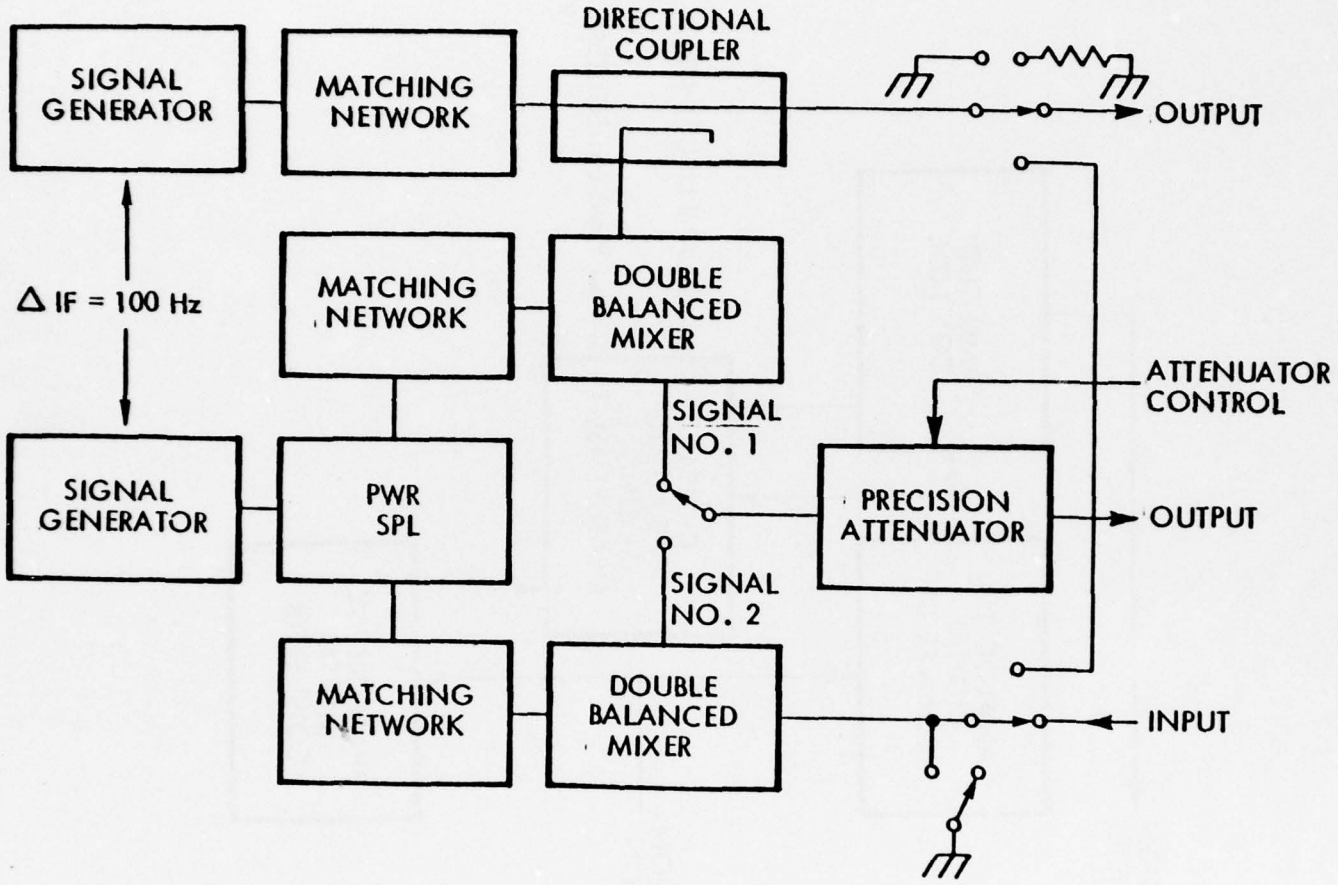


Figure 36. CAL1 RF Subsystem, Block Diagram

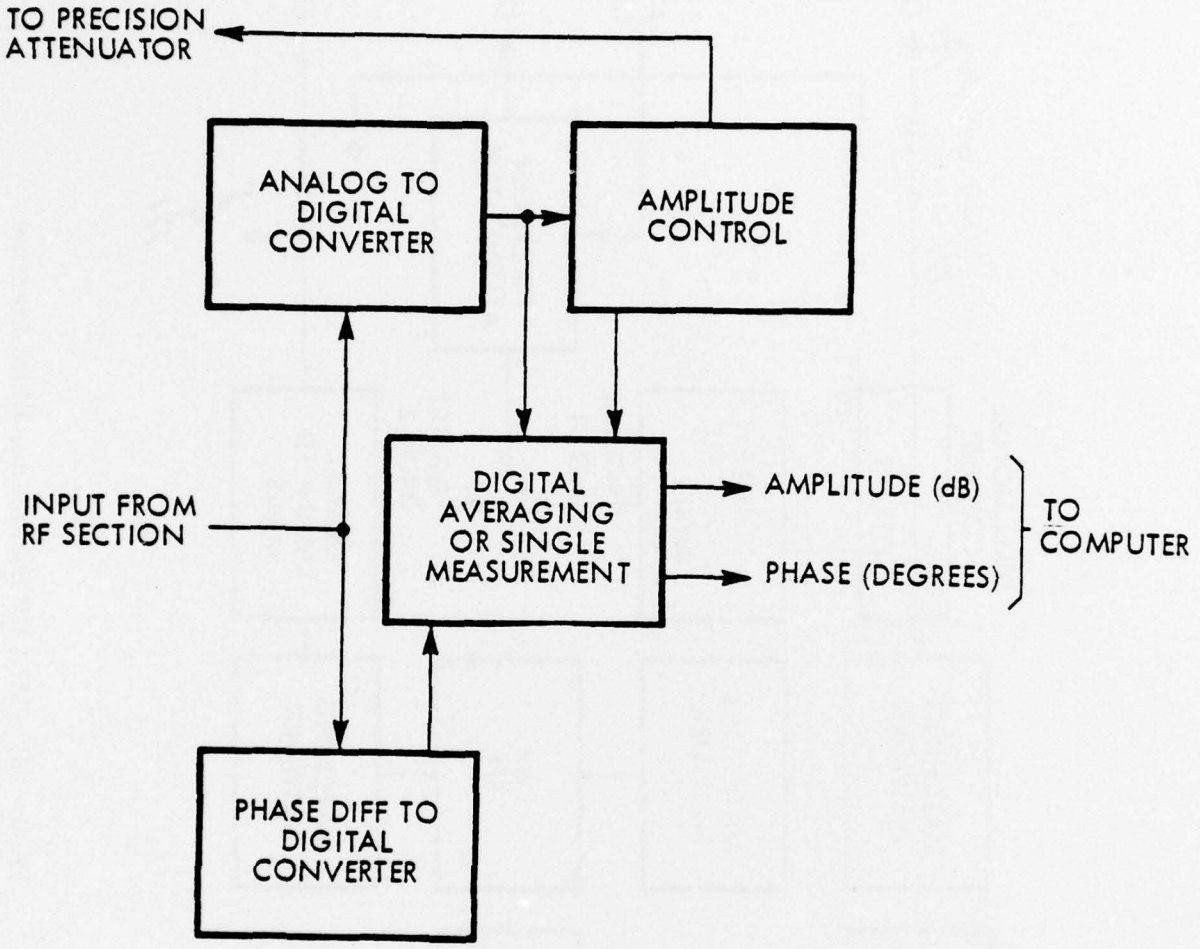


Figure 37. CAL1 Measuring Subsystem, Block Diagram

2.4.2.3.2.1 Amplitude

The amplitude measurement and control is derived using the analog-to-digital converter. In normal use, the ADC (12 bits) would have a dynamic range of 72 dB. With a signal level at maximum, the resolution would be 0.00212 dB [20 log (4094/4095)]. However, with a signal level at minimum, the resolution would be 6.02 dB. Therefore, to use the ADC and obtain a given resolution, the output of the ADC is used in a feedback loop to control the precision attenuator. This loop maintains the voltage into the ADC at a level within ±0.16 dB of a given reference.

Given an ADC with N bits and E_{max} as the maximum voltage, it is desired to find a voltage such that deviations about this point are linear in dB (or nearly so); that is, each step represents a fixed dB value. This voltage (ADC digital value) is found as follows.

The incremental voltage step (E_s) for the given ADC would be

$$E_s = \frac{E_{max}}{2^N} \tag{11}$$

There is a voltage E_{ref} that is represented by the K^{th} step in range of 0 - (2^N -1) steps; thus,

$$E_{ref} = K_{ref} E_s \tag{12}$$

There is another voltage E_v that is represented by an offset (J) relative (plus or minus) from K_{ref} such that

$$E_v = E_{ref} + JE_s \tag{13}$$

The dB change (ΔdB) from E_{ref} to E_v is found by

$$\Delta dB = 20 \log \left(\frac{E_v}{E_{ref}} \right) \tag{14}$$

Substitution of Equations (11), (12), and (13) into Equation (14) yields

$$\Delta dB = 20 \log \left(\frac{J}{K_{ref}} + 1 \right) \tag{15}$$

The minimum value or incremental value (IdB) for ΔdB is obtained for (I=1) a single step removed. Therefore,

$$I_{dB} = 20 \log \left(\frac{1}{K_{ref}} + 1 \right) \tag{16}$$

Solving Equation (16) for K_{ref} gives

$$K_{ref} = \frac{1}{\left[10 \left(\frac{I_{dB}}{20} \right) - 1 \right]} \tag{17}$$

or

$$= \frac{1}{\text{Antilog} \left(\frac{I_{dB}}{20} \right) - 1}$$

Since K_{ref} must be integer, the value found by Equation (17) must be rounded to the nearest integer value. Therefore, using Equation (17), a reference value (K) for any desired resolution (I_{dB}) can be found; the only constraint is, of course,

$$K_{ref} < 2^N - 1 \tag{18}$$

The ADC used in the amplitude measuring subsystem was 12 bits (including sign) and E_{max} was ±5 volts. The resolution chosen was 0.0025 dB, resulting in a value for K_{ref} = 3474. This results in an input value of 8.4814 volts, which exceeds the 5-volt peak value available. Therefore, it was decided to use the E_{ref} as a peak-to-peak voltage. Thus, an E_{max} and an E_{min} are found during each measurement

period. The ADC value for these levels are subtracted to obtain the value for J such that

$$J = (K_{max} - K_{min}) - K_{ref} \tag{19}$$

This technique has the added value of eliminating the effect of any dc bias from the amplifiers. The dB difference from the reference is approximately found as

$$\Delta dB \approx 0.0025J \text{ (dB)} \tag{20}$$

This approximation is used to control the feedback to the precision attenuator, which has values in a binary series (in dB) starting at 0.16 dB and contains nine bits, for a total attenuation of 81.76 dB.

The information passed to the computer as an amplitude value is a 16-bit word. The word contains nine bits of attenuator control and seven bits representing the offset (J), which is converted in the computer to a dB value using Equation (15).

The sample rate used for the ADC was 72 kHz, which provides samples of the 100-Hz signal at 0.5-degree intervals, which results in the selection of E_{max} and E_{min} well within the 0.0025 dB resolution of the ADC (±1.37 degrees around the peak values).

2.4.2.3.2.1 Phase

To convert the phase of the 100-Hz signal to a digital value, the following technique was used. Essentially, three signals are generated:

a. a 100 Hz signal from the RF section
b. a 100 Hz reference signal
c. a 3.6 MHz clock pulse.

The 100-Hz reference signal is used to clear a 16-bit accumulator and then enable a gate, which allows the accumulator to start counting the 3.6 MHz clock pulses. The 100-Hz signal is used to disable the gate, which stops the accumulator counting. The count contained in the accumulator then represents the phase difference between

the signal and the reference. The measurement period is one cycle of the 100 Hz, or 10 ms; thus, the maximum count will be

$$\text{Max Count} = 3.6\,(10^{6})(10^{-2}) - 1 = 35999$$

Then, the accumulator count can range from 0 to maximum, which represents 0 to 359.99 degrees, resulting in a digital conversion to within ±0.005 degree.

To assure this precision, all synthesized signals and oscillators used were derived from the station standard, which was an atomic clock.

2.4.2.3.2.3 Averaging

To improve the precision of the measurements (increase the signal-to-noise ratio), the measured values could be averaged over 64 measurement periods (640 ms). The average was found by accumulating the values for 64 periods and then dividing the sum by 64 before transfer to the computer.

The amplitude values were determined directly, however, the phase values were treated slightly differently.

The phase converter transfer characteristics are shown in Figure 38. The converter inherently produces the solid-line transfer curve. In order to accurately average the phase in the region near zero phase (where the 64 samples could conceivably jump between small values near zero and large values near 359.99 degrees), the curve was extended along one of the two dashed sections of the curve. This extension was accomplished by using the first sample of the period as a reference. If any succeeding sample value differed from the reference by greater than 180 degrees, a value of 360 degrees was either added or subtracted from the sample, thereby defining which extension was being used. Thus, the average could have a negative value or a value greater than 360 degrees; in either case, the computer program could easily detect this and either add or subtract appropriately the 360 degrees required to bring the value within range.

It should be noted that while provision for averaging the measurements was available, it was a computer-selected option. Thus, if the higher precision was not required, the system could return a value using a single measurement period. This results in higher measurement rates where lower precision is acceptable.

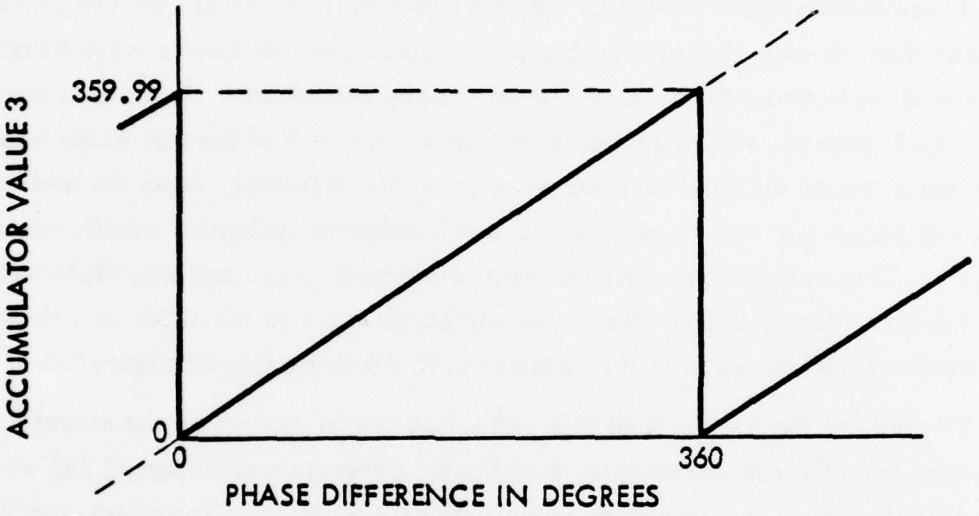


Figure 38. Phase Converter Transfer Characteristic

2.4.2.4 Preliminary Tests

Preliminary tests, using an HP vector voltmeter and a breadboard RF section were made to test the CALI concepts. The breadboards were such that switching of shorts, loads and cable, and data recording were manual. The test setup is shown in Figure 39.

The configuration was changed (i.e., the extra directional coupler in the reference channel) to allow the vector voltmeter to be zero-centered for calibration purposes.

These tests showed the CALI concept was viable; however, errors or variations greater than the expected errors due to measurement resolutions were found. These variations were found to be caused by secondary reflections. In the test configuration shown in Figure 39, the reflection coefficient at the end of the test cable was known when the short or the load resistor were present. However, when the test cable and the cable under test were connected, a small unknown reflection coefficient was present. This reflection coefficient caused periodic phase and amplitude variations as a function of frequency. The phase variations were in the order of 7 degrees and amplitude variations were in the order of 0.75 dB using this configuration.

To confirm the presence of this reflection and to determine the extent of the influence upon the measurements, a computer program was designed and written to calculate the phase and amplitude variations as a function of frequency and VSWR. An attempt was made to duplicate the measured values. The program was capable of duplicating the measured values obtained using the vector voltmeter. However, it also quickly illustrated the necessity of not only including multiple reflections (even to match the vector voltmeter), but additional reflections to use measurements of the precision available from the CALI measuring system.

The internal reflections used are illustrated in Figure 40. The total energy reflected back to the directional coupler is given by

$$E_{RT} = E_{Rd} + \sum_{1}^{k} E_{R(k)} \qquad (21)$$

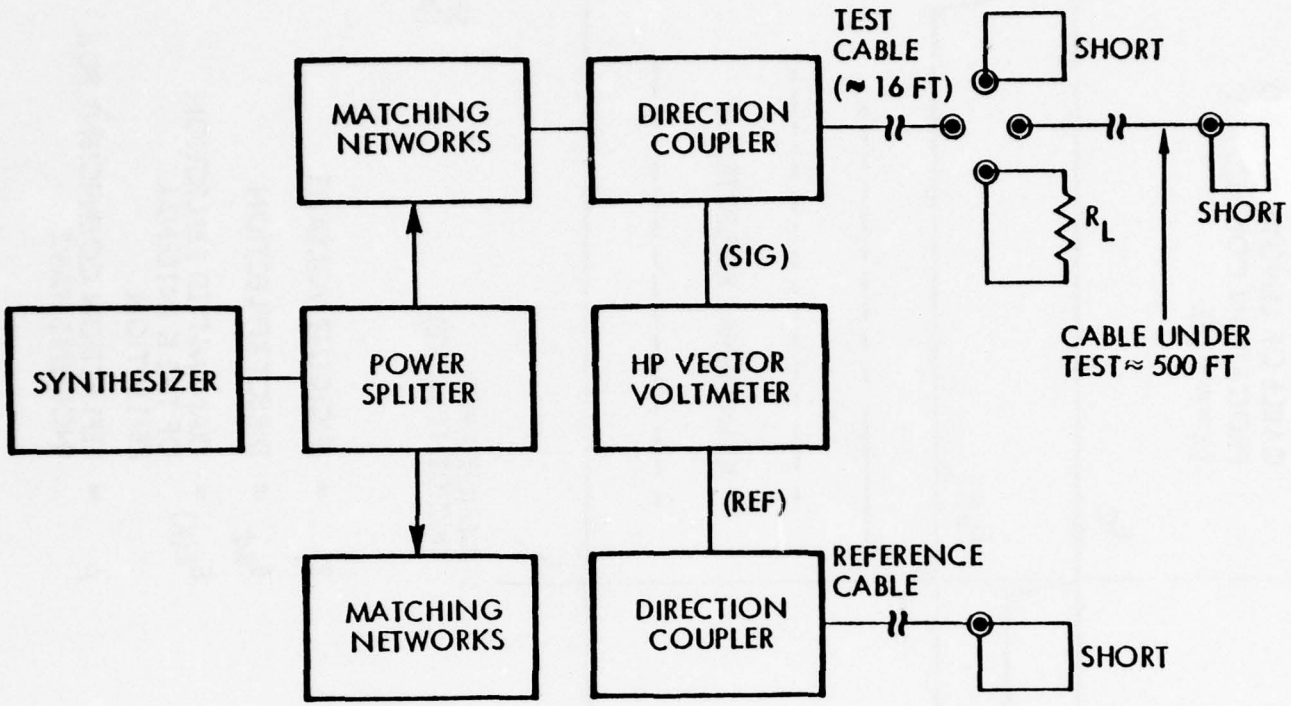
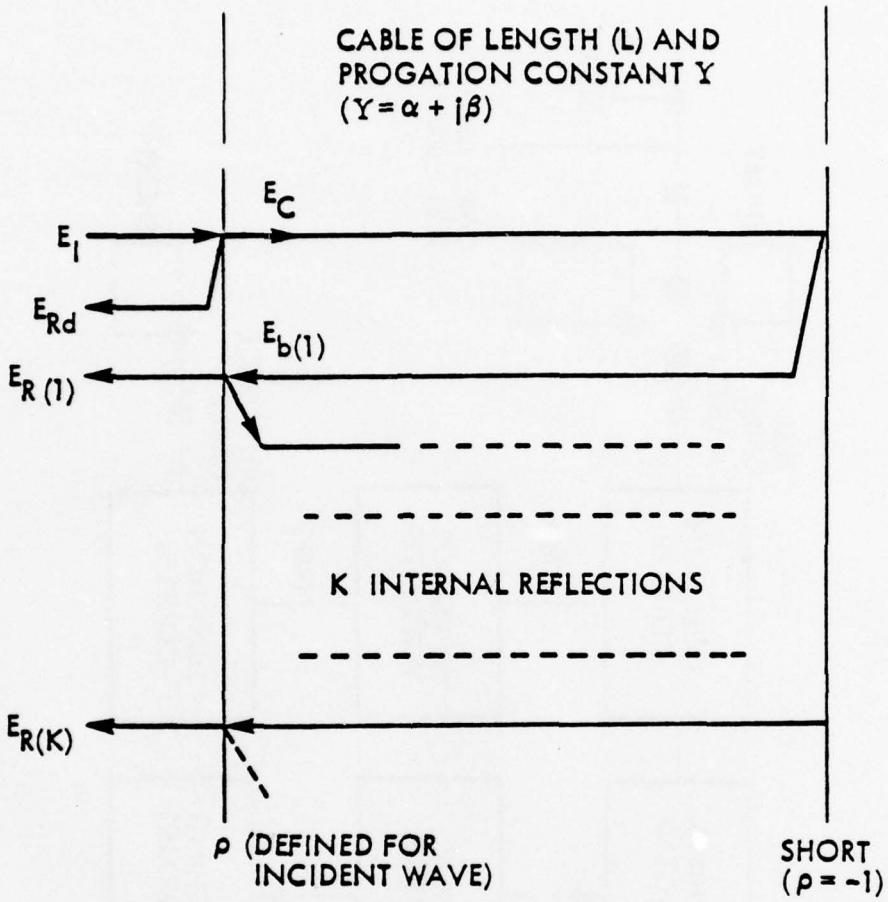


Figure 39. Test Configuration for Preliminary CAL1 Tests



E_I = INCIDENT VOLTAGE

E_{Rd} = DIRECT REFLECTION

$E_{R(K)}$ = TRANSMITTED REFLECTION OF THE K INTERNAL REFLECTION

ρ = REFLECTION COEFFICIENT FOR INCIDENT WAVE

Figure 40. Internal Reflections for One Secondary Reflection Point

where

$$E_{Rd} = \rho E_I \text{ , and}$$
$$E_{R(k)} = \rho^{(k-1)} (1 - \rho^2) E_I e^{-2k\lambda L}$$

If the short circuit were removed and replaced by ρ_1, the value for $E_{R(k)}$ becomes

$$E_{R(k)} = (-1)^{k-1} \rho_1^{\,k} \rho^{(k-1)} (1 - \rho^2) E_I e^{-2k\lambda L} \tag{22}$$

for use in Equation (21).

In addition, it can be similarly shown that the transmission through the cable becomes the incident signal for the next section and is found by Equation (23):

$$E_{I(1)} = E_I (\rho + 1) (\rho_1 + 1) e^{-\lambda L} \left\{ 1 + (-1)^{k-1} (\rho\rho_1)^{k-1} e^{-2(k-1)\lambda L} \right\} \tag{23}$$

The addition of more than one secondary reflection point considerably increases the complexity of the returning signal, as illustrated in Figure 41. However, for the section between two small reflection coefficient values, the multiple reflections diminish rapidly. (The magnitude of the vector must be 85 dB or greater below the summation before the reflections can be neglected, if ±0.005 degree accuracy is expected.)

The greatest task is to characterize the Dexter CALI system in terms of secondary reflection points in order to evaluate the phase and amplitude of the trimming units in the system. The magnitude of this task is easily seen by inspection of Figure 41 and determining the many possible reflection points. This characterization was not initiated prior to the project termination.

One difficulty in the CALI system was uncovered prior to termination. The relay selected by the Project Engineer for the shorting relay in the field was not an RF relay; therefore, it had the characteristics of inserting a small inductance instead of a short. An analysis was not made to determine if this relay could be tolerated, even if adequately described.

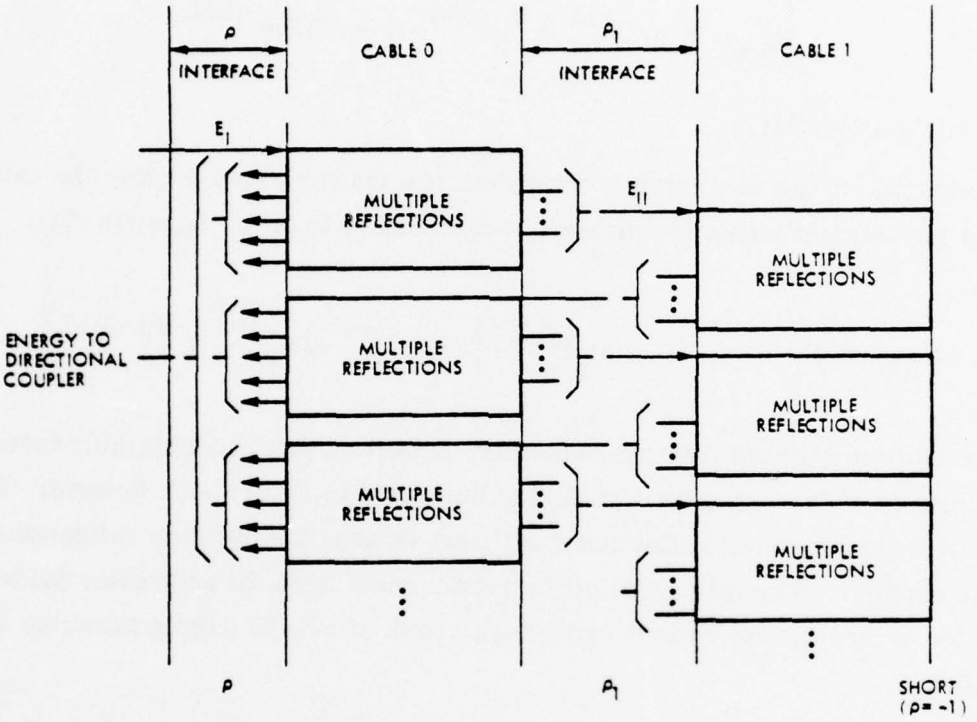


Figure 41. Internal Reflections for Two Secondary Reflection Points

2.5 ARRAY PERFORMANCE MODEL

The array performance model program is designed to accept, if available, the element median pattern and the RMS errors in phase and amplitude as a function of azimuth obtained from the flight check data and reduction program.

With these data, the program calculates the required patterns with and without errors for the requested array and/or bit parameters. The program accepts sets of requested calculations, storing each requested parameter set on disk storage.

The calculated patterns and associated data are recorded sequentially on magnetic tape.

All listings for this program were delivered in place.

2.5.1 Array Parameters

There are six array parameters available for input selection by the operator.

a. Number of rows and row spacing

b. Number of columns and column spacing

c. Sidelobe weighting (Taylor or uniform) and value

d. Array azimuthal steering angle

e. Array vertical steering angle

f. Array focal length infinite or finite

   If finite:

   (1) include or exclude element pattern (if available)

   (2) use element orbit parameters (if available) for

      (a) elevation and

      (b) range

   (3) select (by operator)

      (a) elevation and

      (b) range

   If infinite: elevation steer angle.

2.5.2 Orbit Parameters

The orbit parameters available for input selection by the operator are

a. Orbit vertical angle

b. Orbit radius infinite or finite. If finite,

   (1) Range and altitude, or

   (2) Element orbit data (if available).

2.5.3 Element Parameters

The element parameters available for input selection by the operator are the following.

a. Is tape available (yes/no)?

b. Sortie number if tape is available.

The data available on the tape (if available) is in groups of five records, with the groups separated by end-of-file marks. Each group contains the information enumerated in Table 12.

2.5.4 Patterns

The patterns available for input selection by the operator are the following.

a. Element pattern used
b. Ideal array factor
c. Ideal array pattern
d. Array factor with beamformer errors
e. Array pattern with beamformer errors
f. Array pattern with element errors
g. Array pattern with element and beamformer errors

Note: The term "factor" is used to denote array factor with no element pattern. The term "pattern" is used to denote array factor combined with the element pattern.

Any combination of patterns may be chosen; however, if a pattern is selected that requires the element pattern and the element pattern is not available, that selection is refused.

TABLE 12. DATA GROUPINGS ON TAPE

Record Number	Contents	Number of Words
1	Header record*	16
2	Azimuthal angles with respect to true north	One per second**
3	Mean element pattern voltage levels	One per second**
4	RMS amplitude error pattern (radians)	One per second**
5	RMS phase error pattern (radians)	One per second**

*All words are fixed-point binary numbers.

Word Number	Meaning
1	Sortie sequence number
2	Sortie type
3	Sortie number
4, 5	Month, day
6	Frequency (Hertz)
7	Aircraft altitude (feet)
8	Array boresight steering direction (1 = 60° T, 2 = 240° T)
9, 10, 11	Sortie start time (hours, minutes, seconds)
12, 13, 14	Sortie stop time (hours, minutes, seconds)
15	Not used
16	Source of aircraft position (1 = PAMS/FPS-16, 2 = Manual voice log)

**One floating-point binary (Sigma-5 format) number per each second of sortie, up to a maximum of 1500 words (i.e., 25 minutes).

2.5.5 Finite Calculations

All finite calculations, either for a focused array or orbits, use a three-dimensional model (element altitudes are assumed zero) that calculates the slant range from the point in space to the individual elements. This range is used to determine amplitude and phase differences for the individual elements.

The phase differences are calculated using free-space velocity, and amplitude differences are calculated using free-space loss. (No attempt was made to modify either velocity or losses due to ground effects.)

This model is also used to determine azimuthal bearings to the individual elements, and these bearings are used to include the element pattern, if required. Since these bearings are individual, the parallax error (which would not occur if a single bearing for all elements were used) is introduced.

2.5.6 Element Error Calculations

The two patterns that require element errors to be included also require the errors to be the same. The errors are calculated during the first pattern, using a random-number generator (using tolerance limits in accordance with the recorded RMS error set). The errors are then used and recorded for use again during the next pattern calculation.

It should be noted that the element errors are a function of azimuth. It is suspected the actual errors are correlated from bearing to bearing; however, no attempt was made to calculate the correlation function in the flight check and reduction program. Therefore, no attempt was made to force any correlation between error sets, even though they are separated by only 0.25 degree in azimuth. Hence, the program calculates a new random set of errors for each azimuth used in the calculations.

2.5.7 Pattern Calculation

The calculation of the pattern was accomplished using Equation (24).

$$E_{(\theta, \phi)} = \frac{1}{NJ} \sum_{n=1}^{n=N} \sum_{j=1}^{j=J} C_{(n,j)} e^{-iS_{(n,j,\theta,\phi)}} \tag{24}$$

where:

$C_{(n,j)}$ = the amplitude coefficient for the port

N = the number of columns

J = the number of rows

$S_{(n,j,\theta,\phi)}$ = the phase angle for the port

Since the values of $C_{(n,j)}$ and $S_{(n,j,\theta,\phi)}$ were modified from the ideal values by the errors or finite calculations, this equation was used for all pattern values.

2.5.8 Flow Diagrams

The general program flow diagram is shown in Figure 42. The detailed flow diagrams for the program and subroutines are shown in Figures 43 through 51.

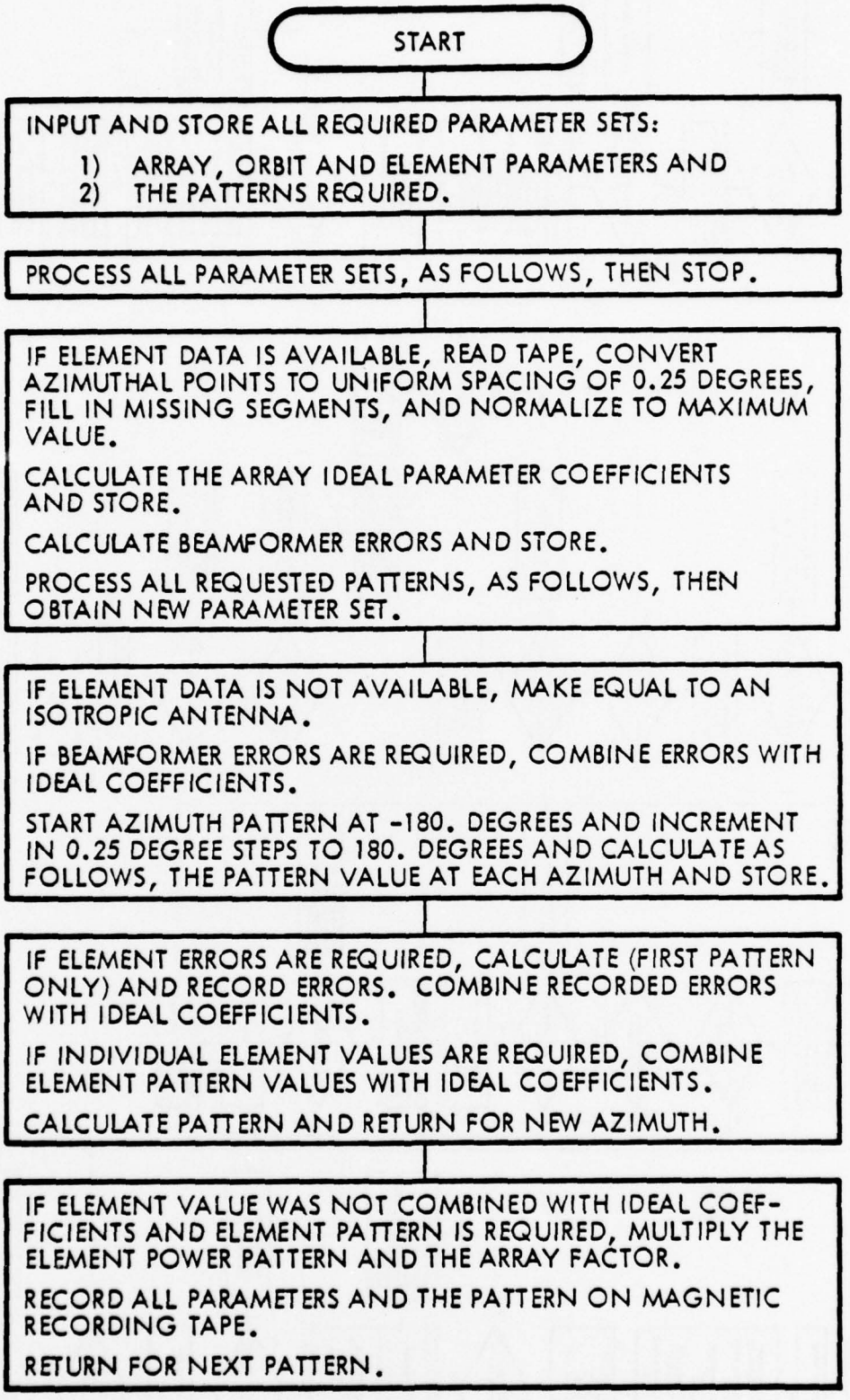


Figure 42. Array Performance Model, General Flow Diagram

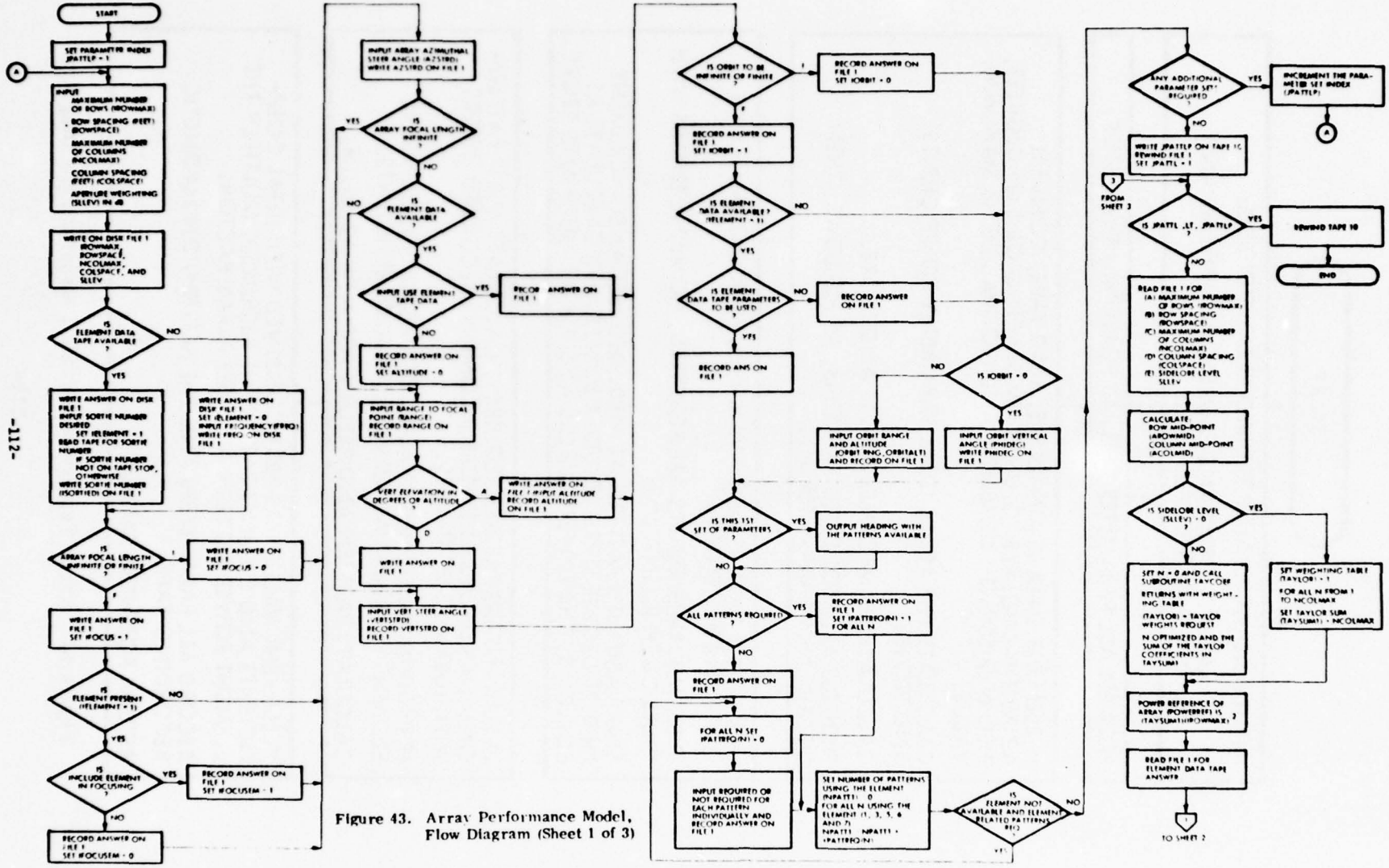


Figure 43. Array Performance Model, Flow Diagram (Sheet 1 of 3)

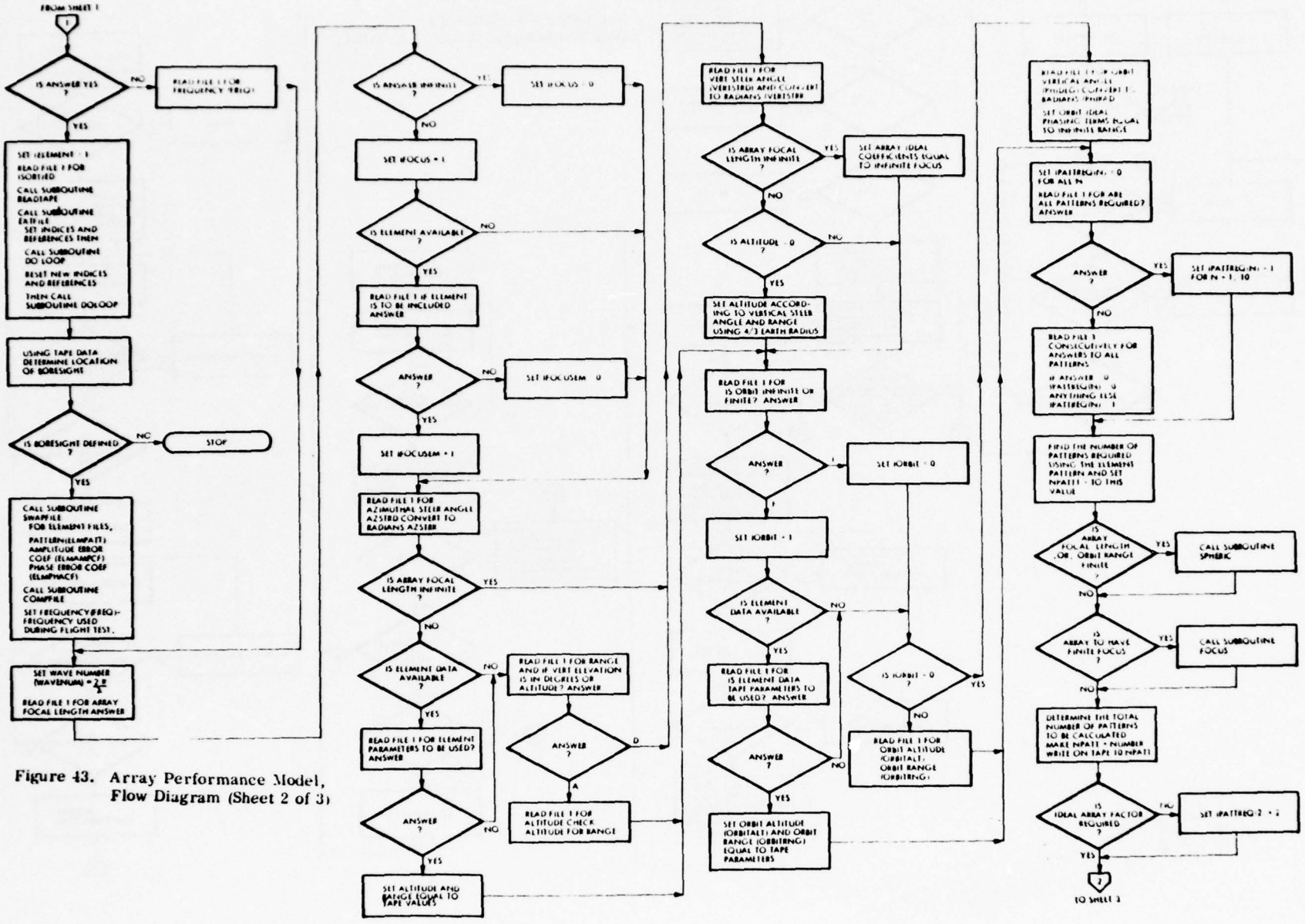


Figure 43. Array Performance Model, Flow Diagram (Sheet 2 of 3)

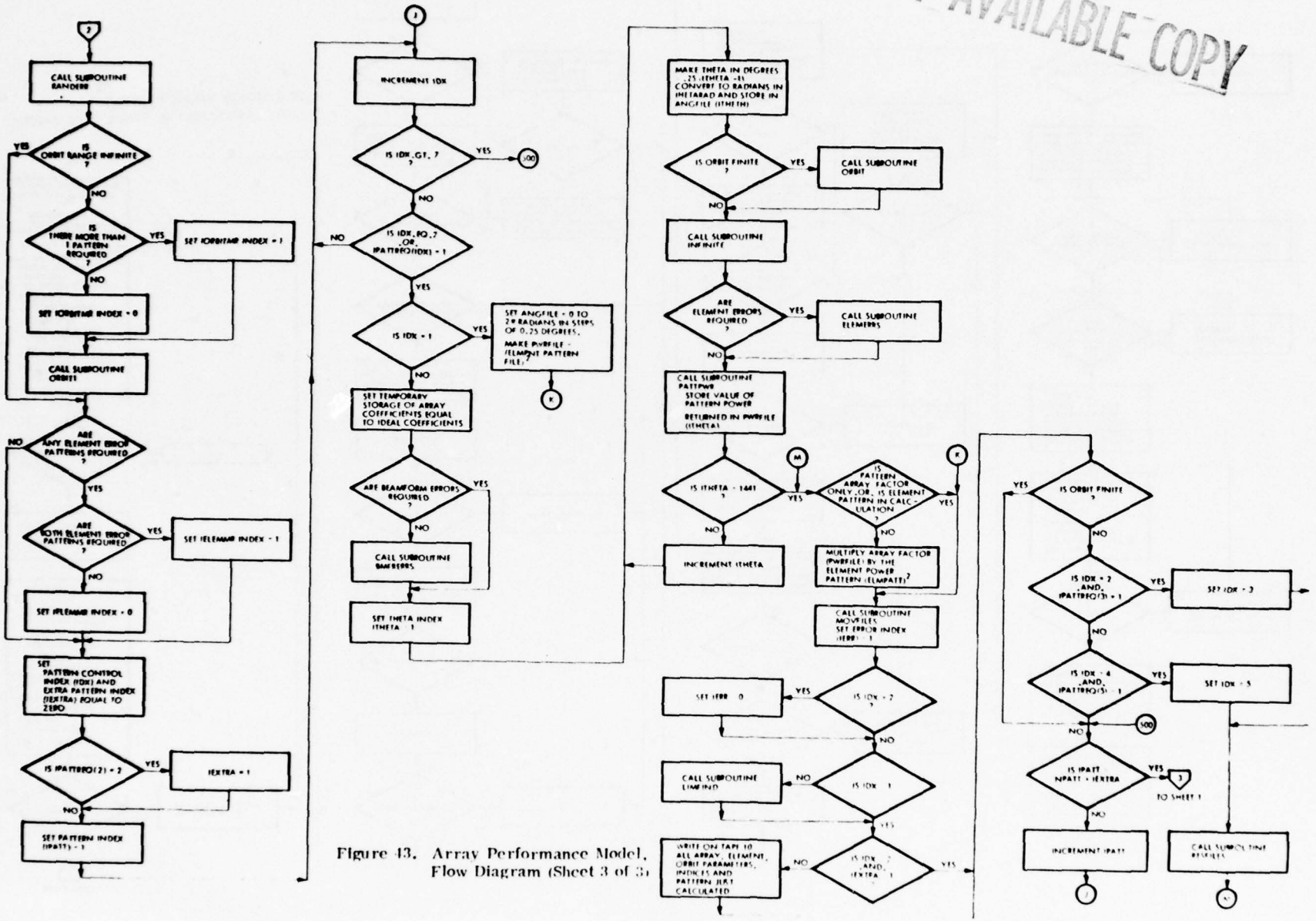


Figure 43. Array Performance Model, Flow Diagram (Sheet 3 of 3)

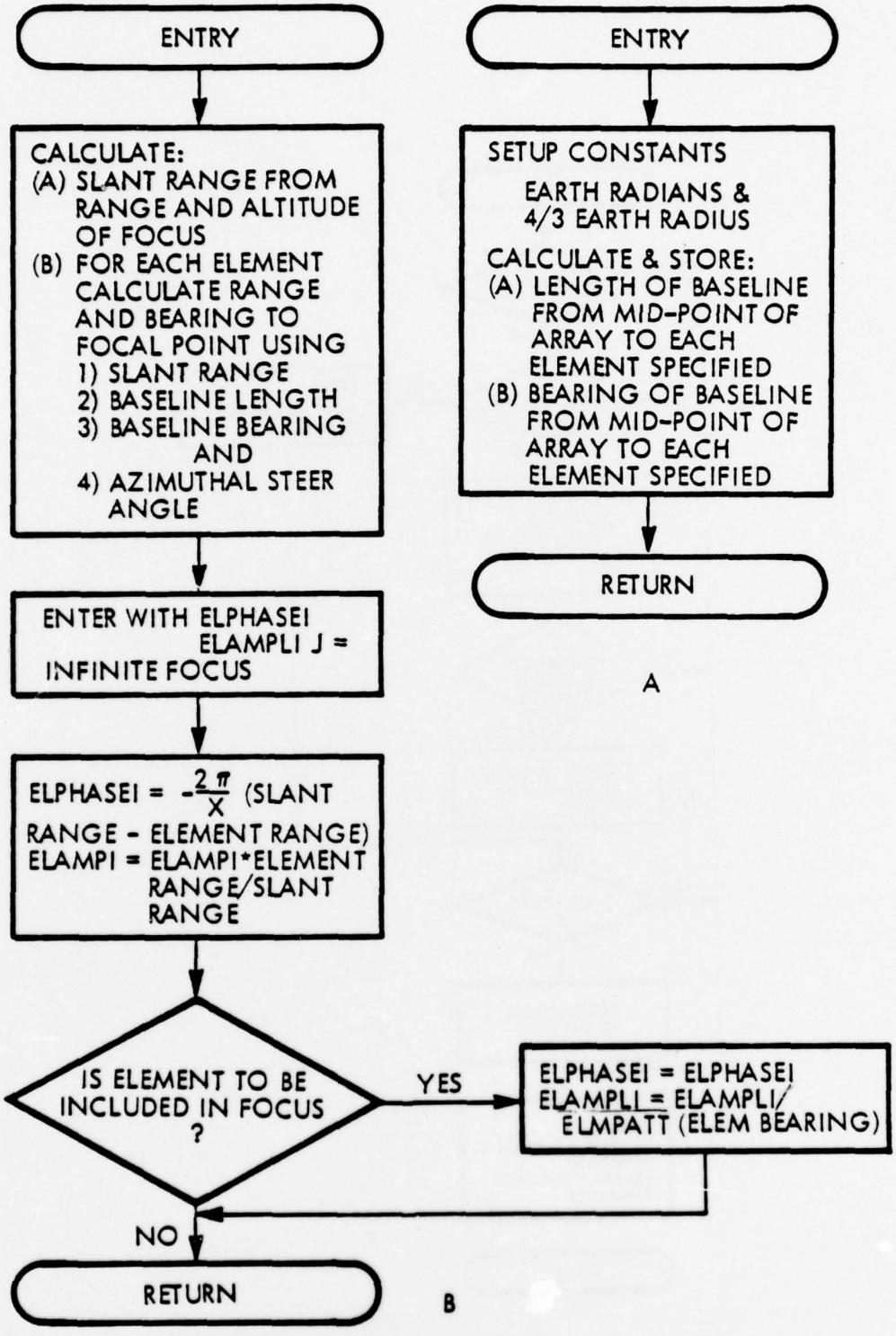


Figure 44. Subroutines FOCUS and SPHERIC, Flow Diagrams

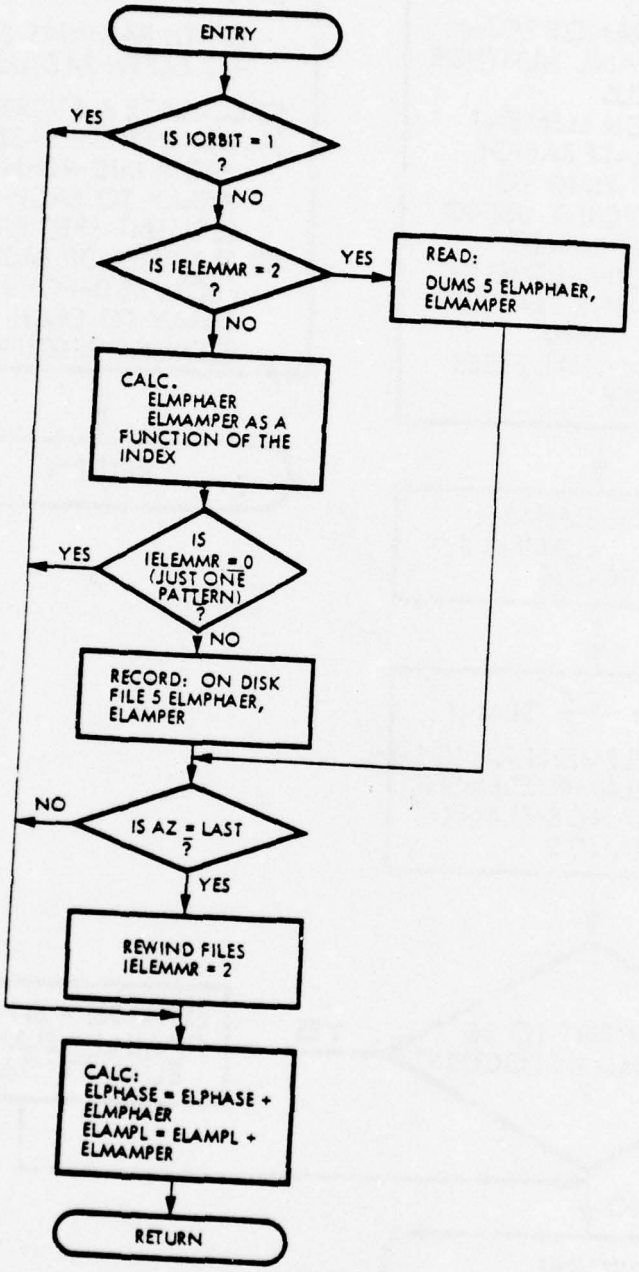


Figure 45. Subroutine ELEMERRS, Flow Diagram

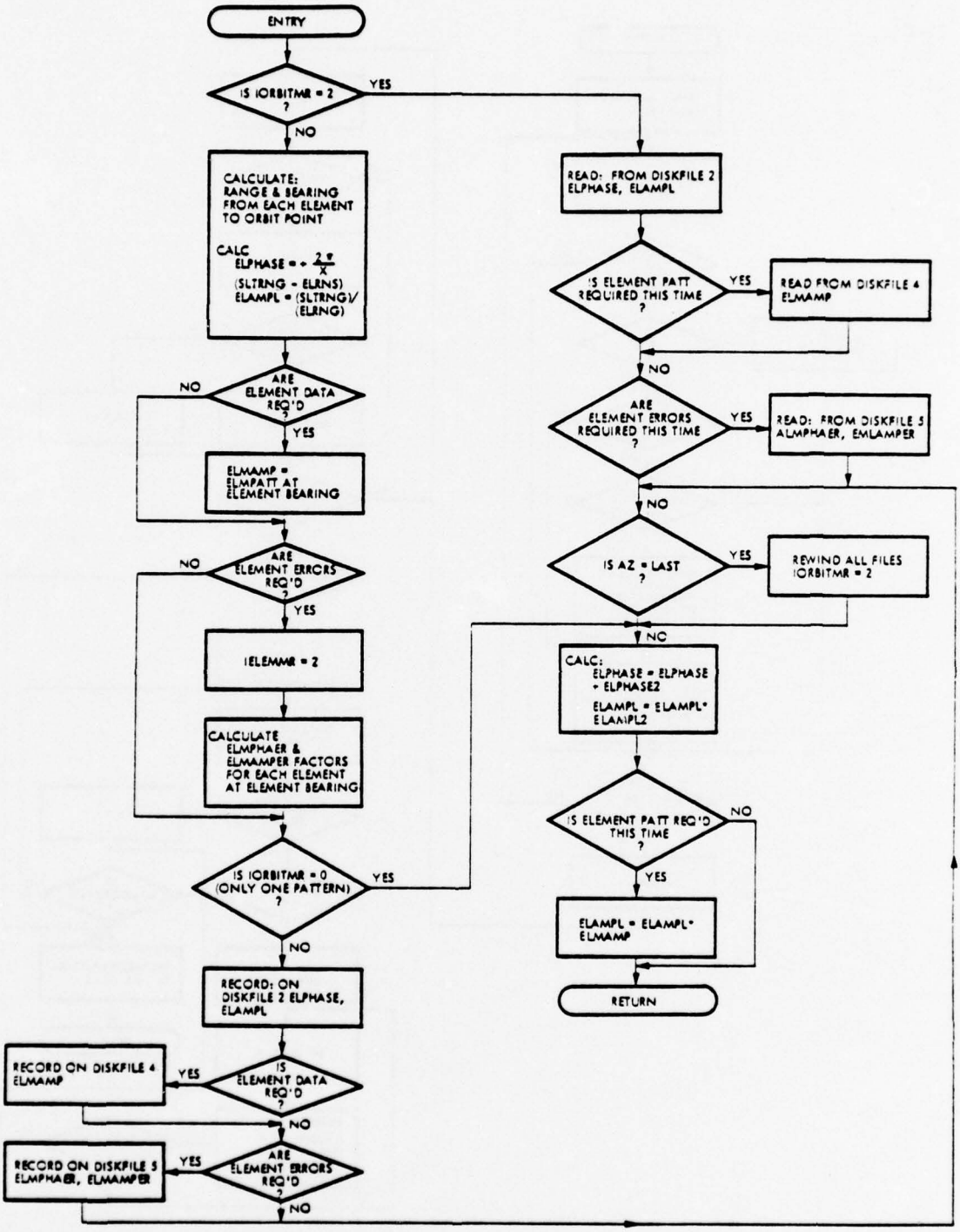


Figure 46. Subroutine ORBIT, Flow Diagram

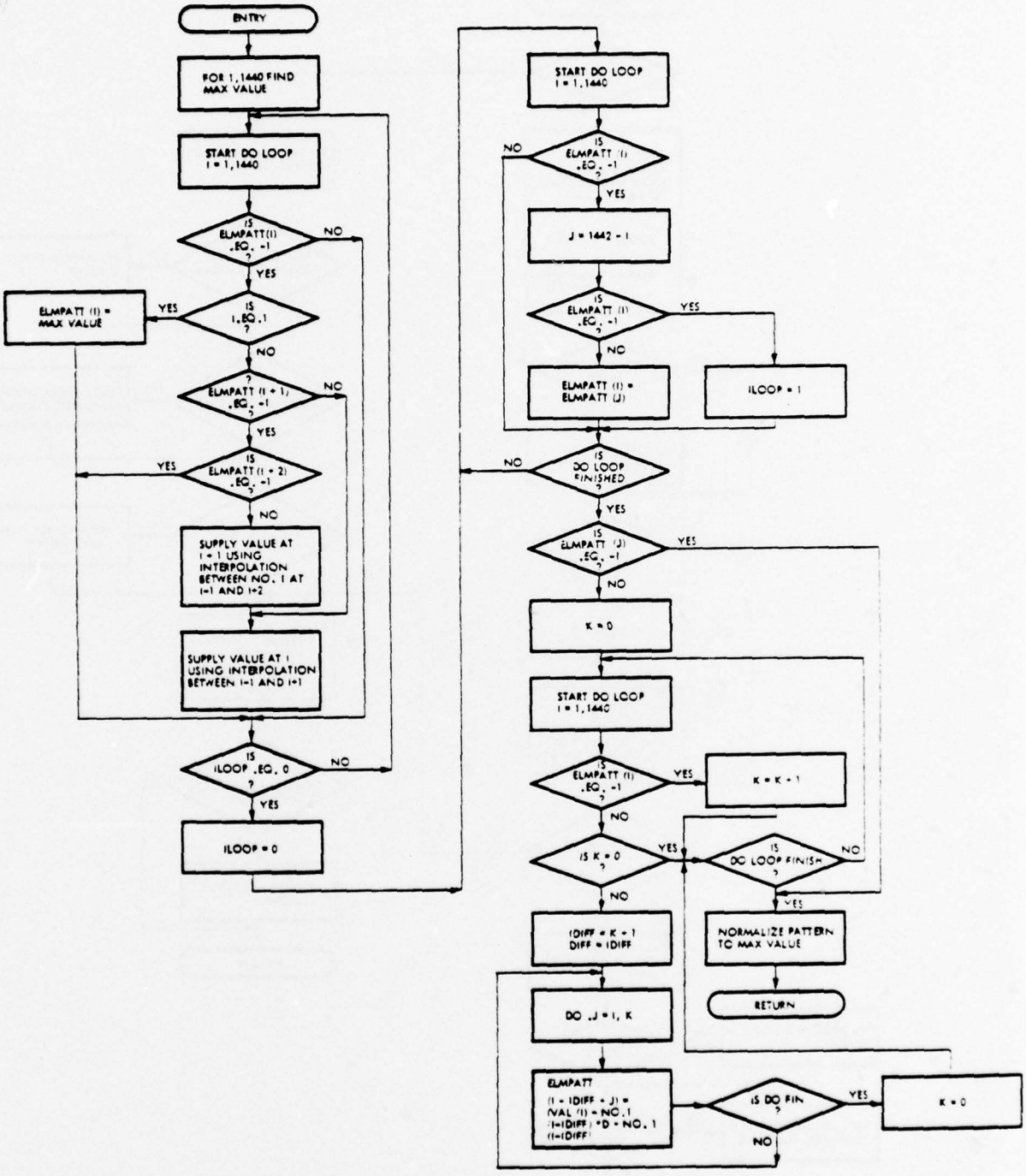
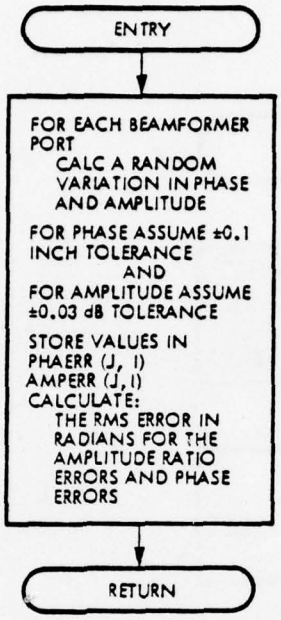
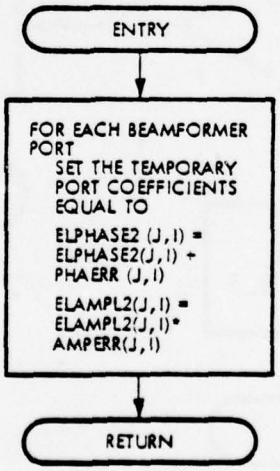


Figure 47. Subroutine COMPFILE, Flow Diagram



(A) SUBROUTINE RANDERR



(B) SUBROUTINE BMFRERRS

Figure 48. Subroutines RANDERR and BMFRERRS, Flow Diagram

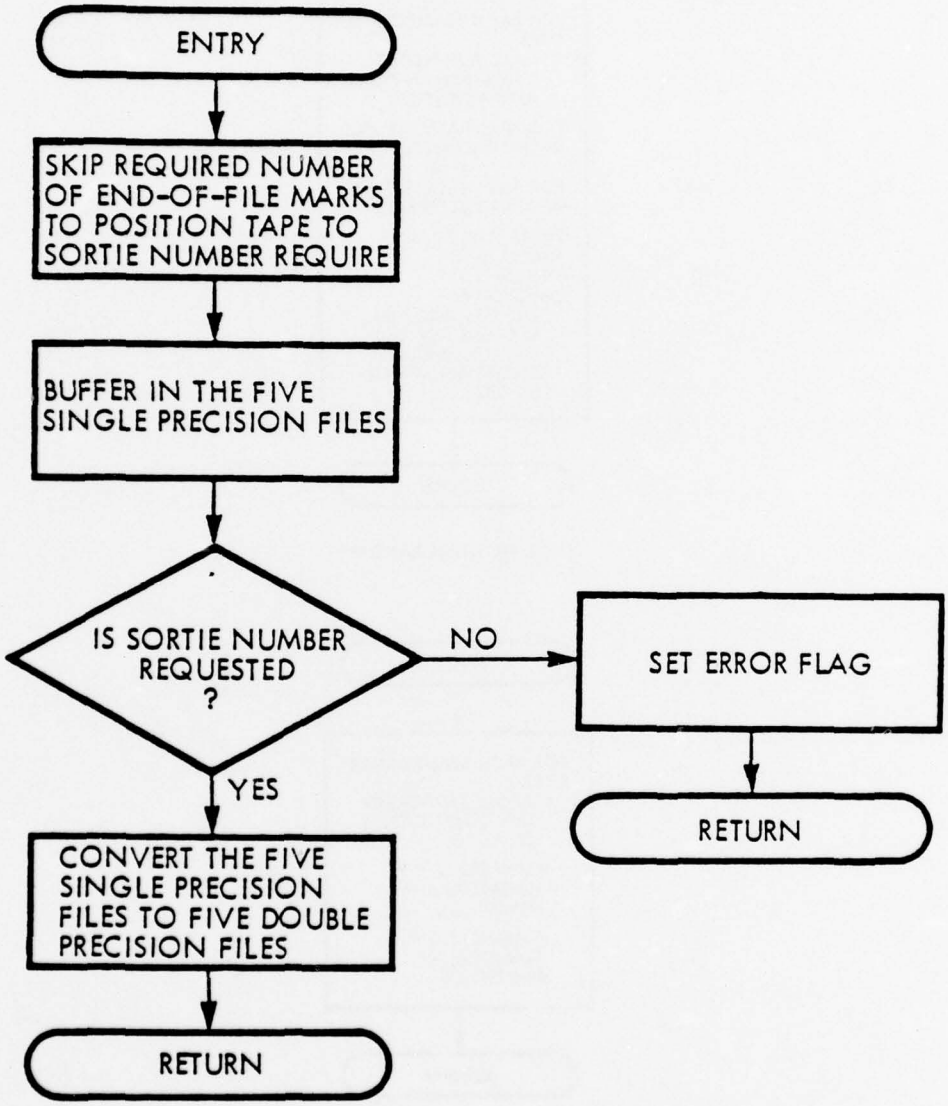


Figure 49. Subroutine READTAPE, Flow Diagram

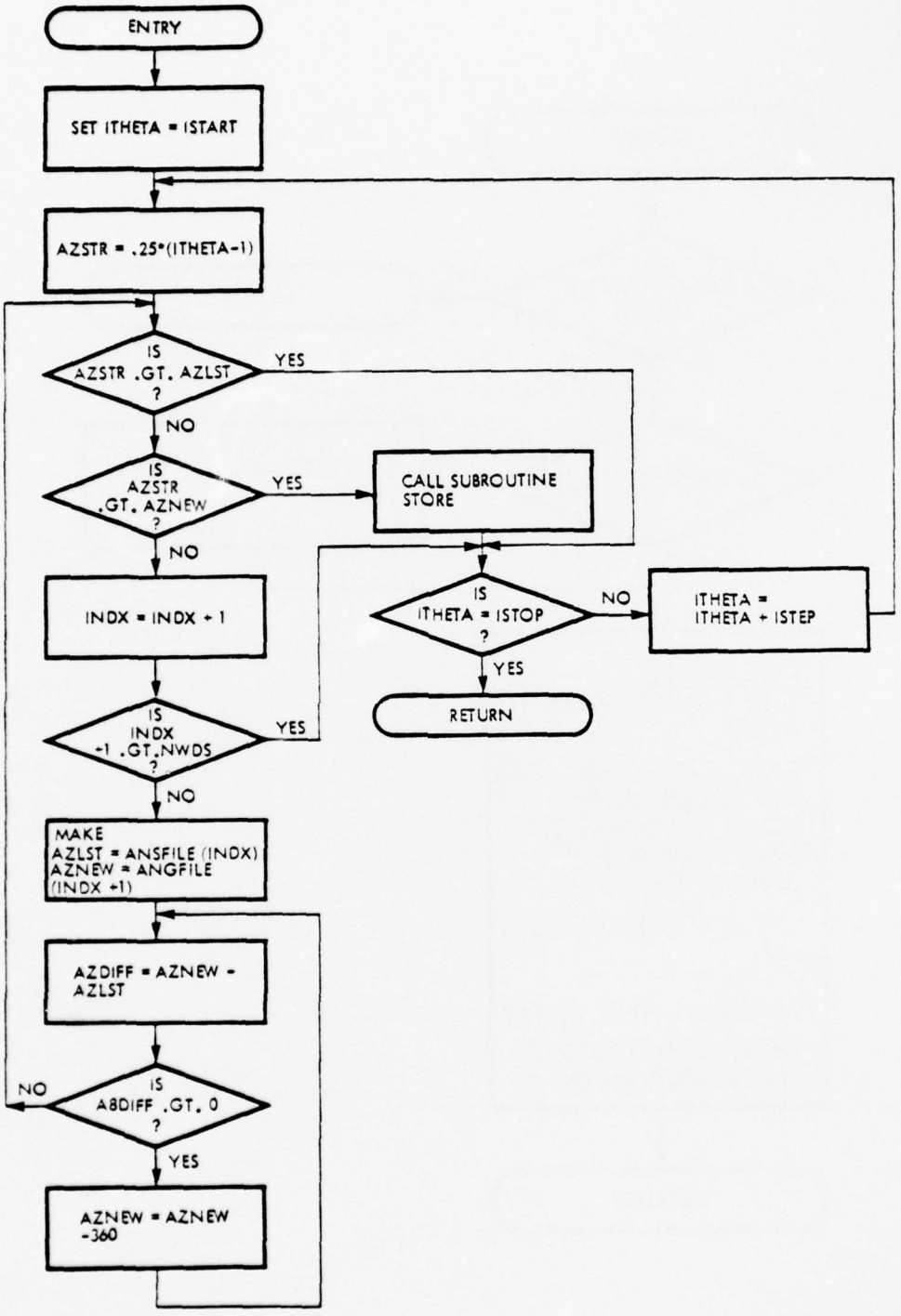


Figure 50. Subroutine DOLOOP, Flow Diagram

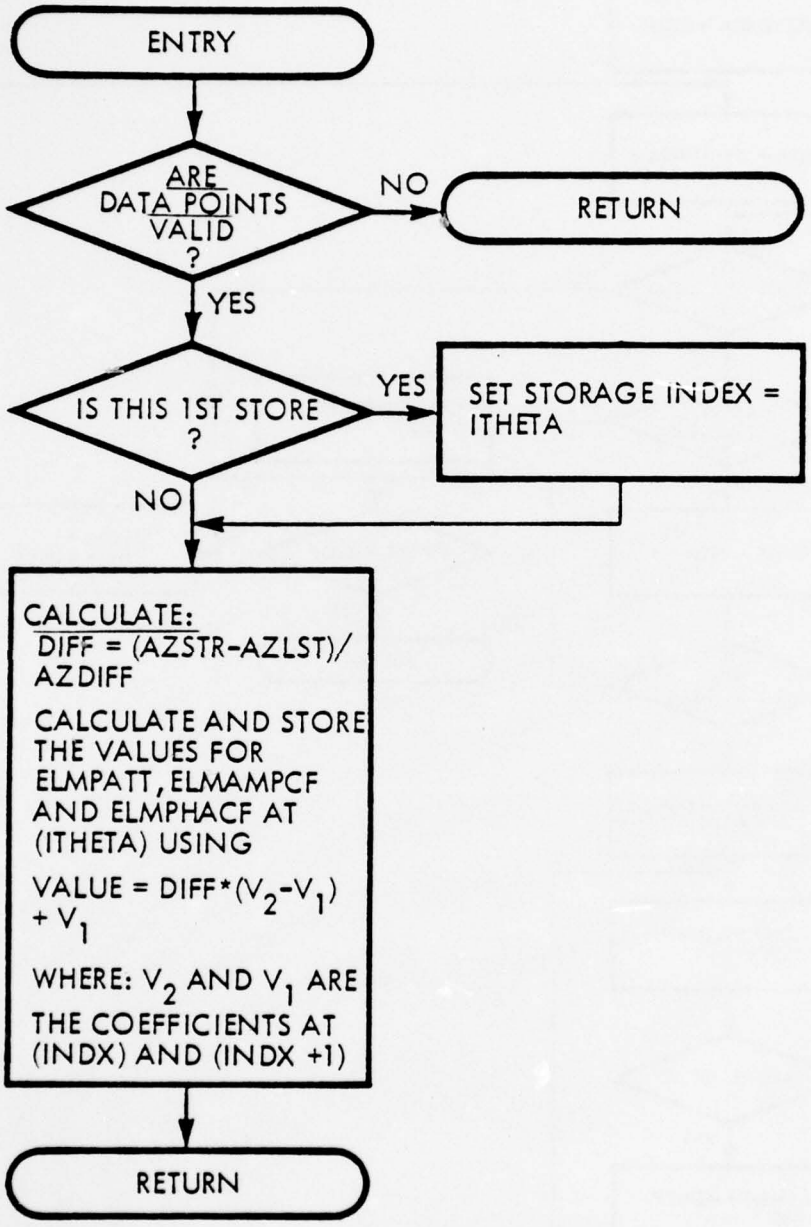


Figure 51. Subroutine STORE, Flow Diagram

2.5.9 Loading and Running Array Performance Model

The Fortran source program and internal subroutines are stored in file ARAYMULT. The external subroutine binary files are stored in files

a. TAYLM:
b. DEXTER:
c. TAPERD:

and any unsatisfied externals are in

d. BILLNOW account under file :SYS

Program MULT, stored in a file of same name, is used for the following functions.

a. Provide proper job card parameters
b. Have listings output to line printer
c. Make all file assignments to disk and tape
d. Compile program ARAYMULT
e. Load the program binary and external files into a permanent load module, named XARAY1

This program (MULT) is required only if changes to the program source file or external subroutine binary files have been made.

The program DEXARAY1 stored in the file of the same name is used for three functions.

a. Provide proper job card parameters
b. Make file assignments to disk and tape
c. Run the load module XARAY1

This program (DEXARAY1) causes program execution to begin, and all operator interface is done through the system console, not the time-share terminal.

To initiate these programs, the user must sign on a time-share terminal with the ID

NAME, 100

make an ASSIGN M:SI, (FILE, DEXARAY1), and enter the job into the BPM system. The BPM system will then request two tapes to be mounted on the nine-track tape units:

a. PATT available from flight test and reduction programs

b. An output tape named ARAY.

2.6 PATTERN PLOTTING

The patterns calculated using the array performance model are plotted using the Fortran source program PATTPLOT stored in the file of the same name. This program performs the following four functions.

a. Provides proper job card parameters

b. Makes assignments for tape and plotter (ARDS)

c. Compiles the Fortran source program

d. Loads compiled binary and external files into a permanent load module named XARAY2

The program is run using program DEXARAY2 stored in the file of the same name. This program performs three functions.

a. Provides proper job card parameters

b. Makes assignments for tape and plotter (ARDS)

c. Runs load module XARAY2

All operator interface is done using the system console, not the time-share terminal.

To use the programs, the user must sign on using the same procedure and ID as given in Paragraph 2.5.9, except the assign is to file DEXARAY2.

The program PATTPLOT reads the tape made using the array performance model and uses this information to

a. draw the axis for the pattern,

b. plot the average sidelobe level,

c. plot pattern, and

d. label plot.

Examples of these plots are shown in Figures 52 through 58. (Note: Error data used was preliminary and does not represent the final values.)

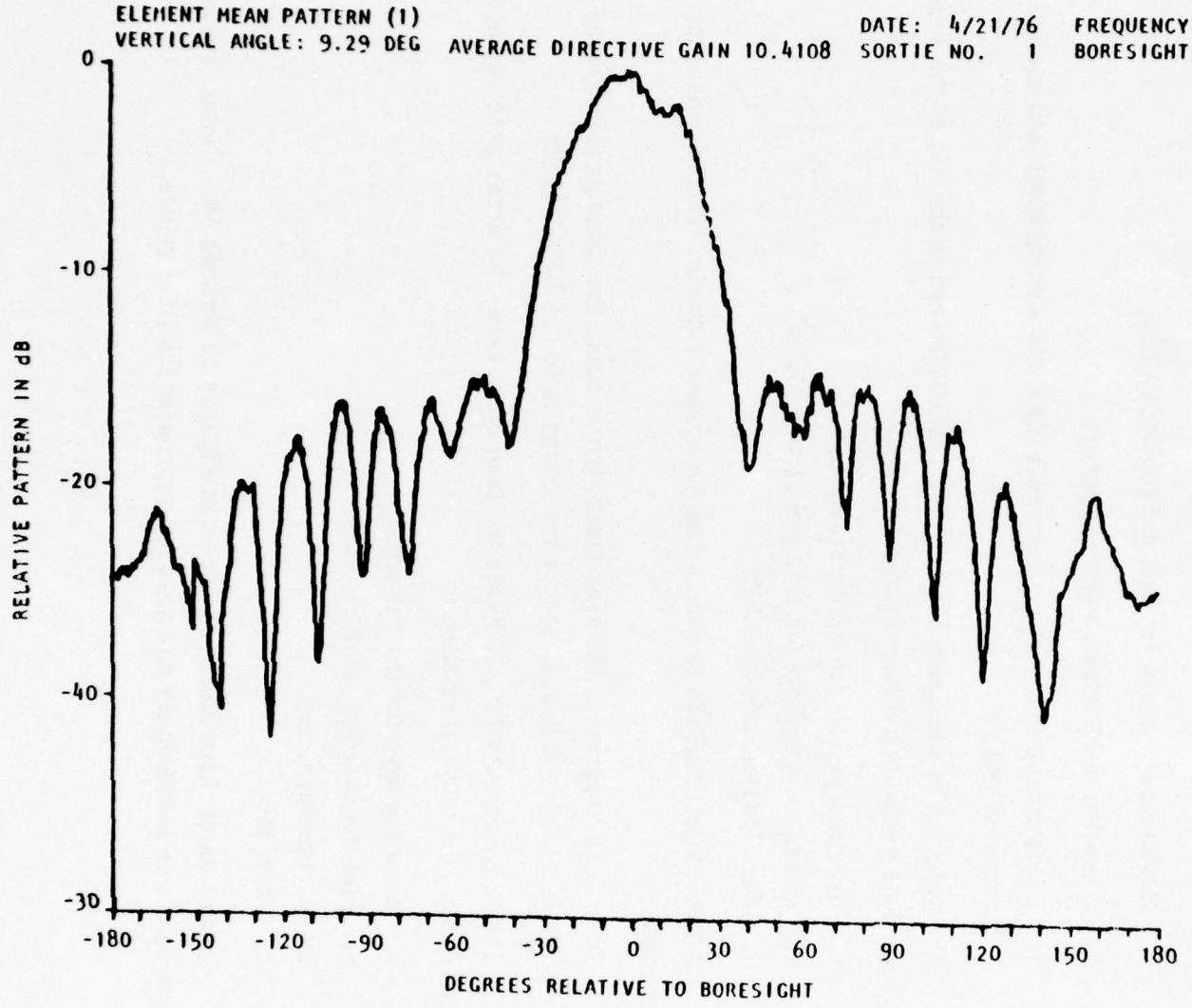


Figure 52. Pattern Plot for Element Mean Pattern (1)

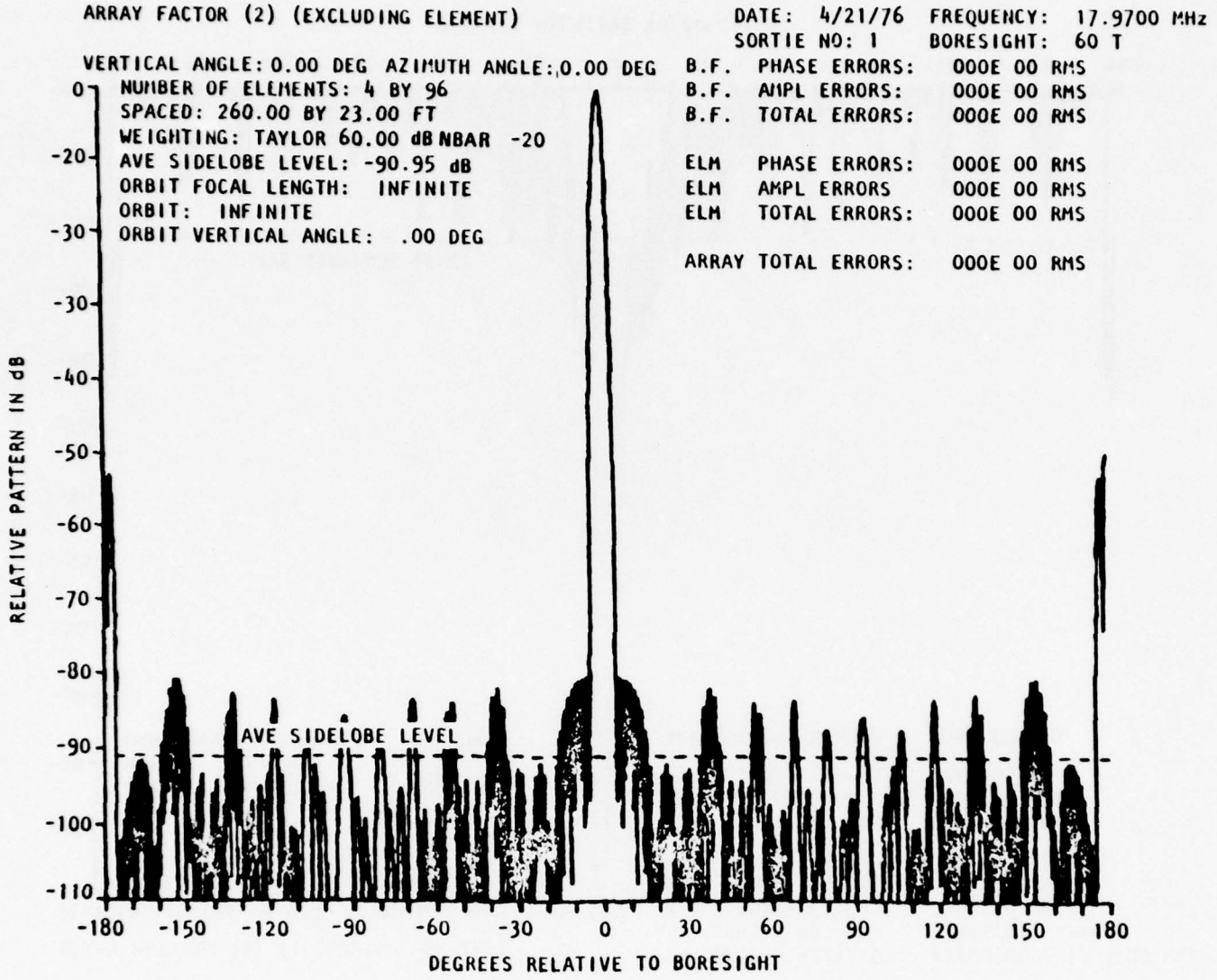


Figure 53. Pattern Plot for Array Factor (2) (Excluding Element)

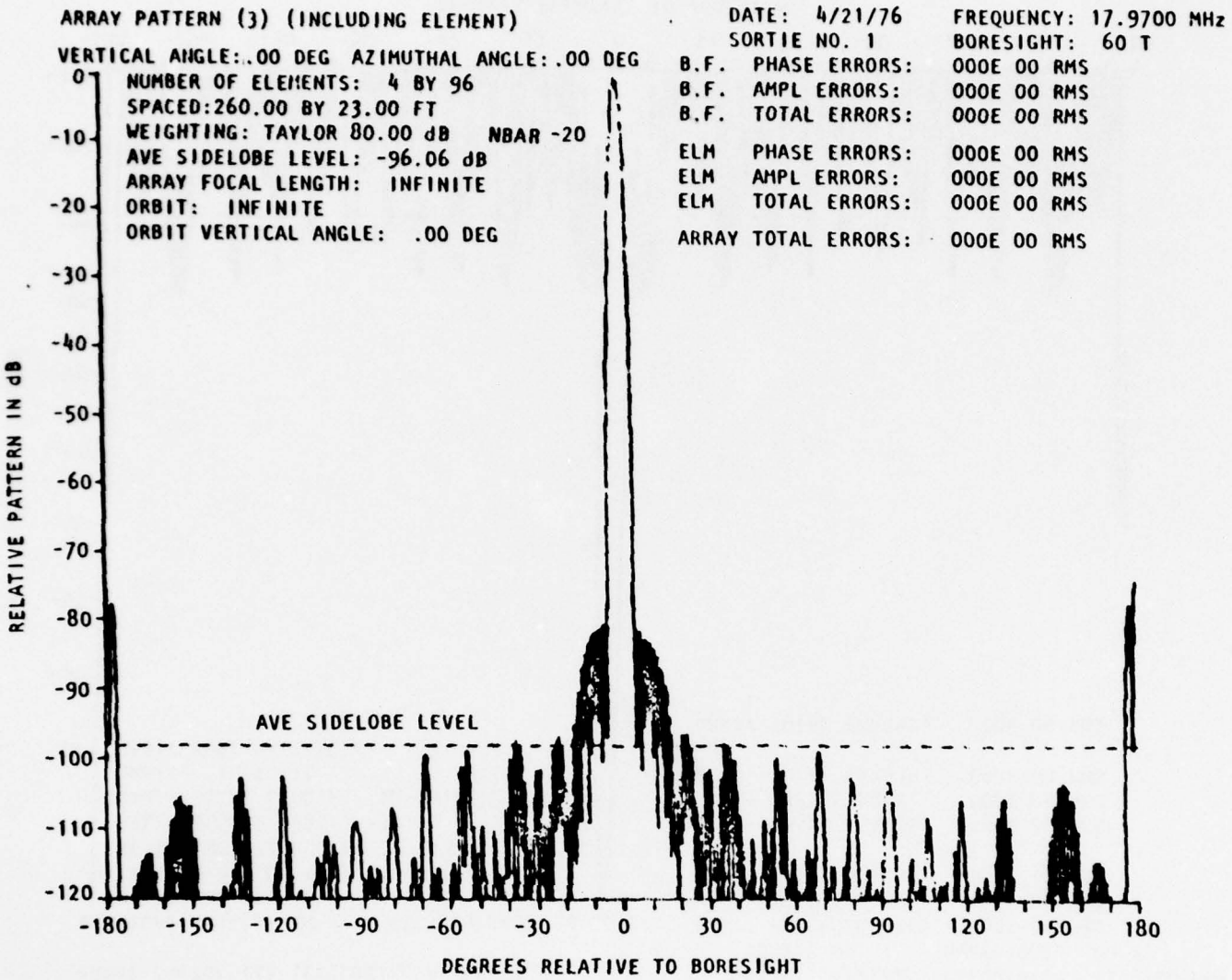


Figure 54. Pattern Plot for Array Pattern (3) (Including Element)

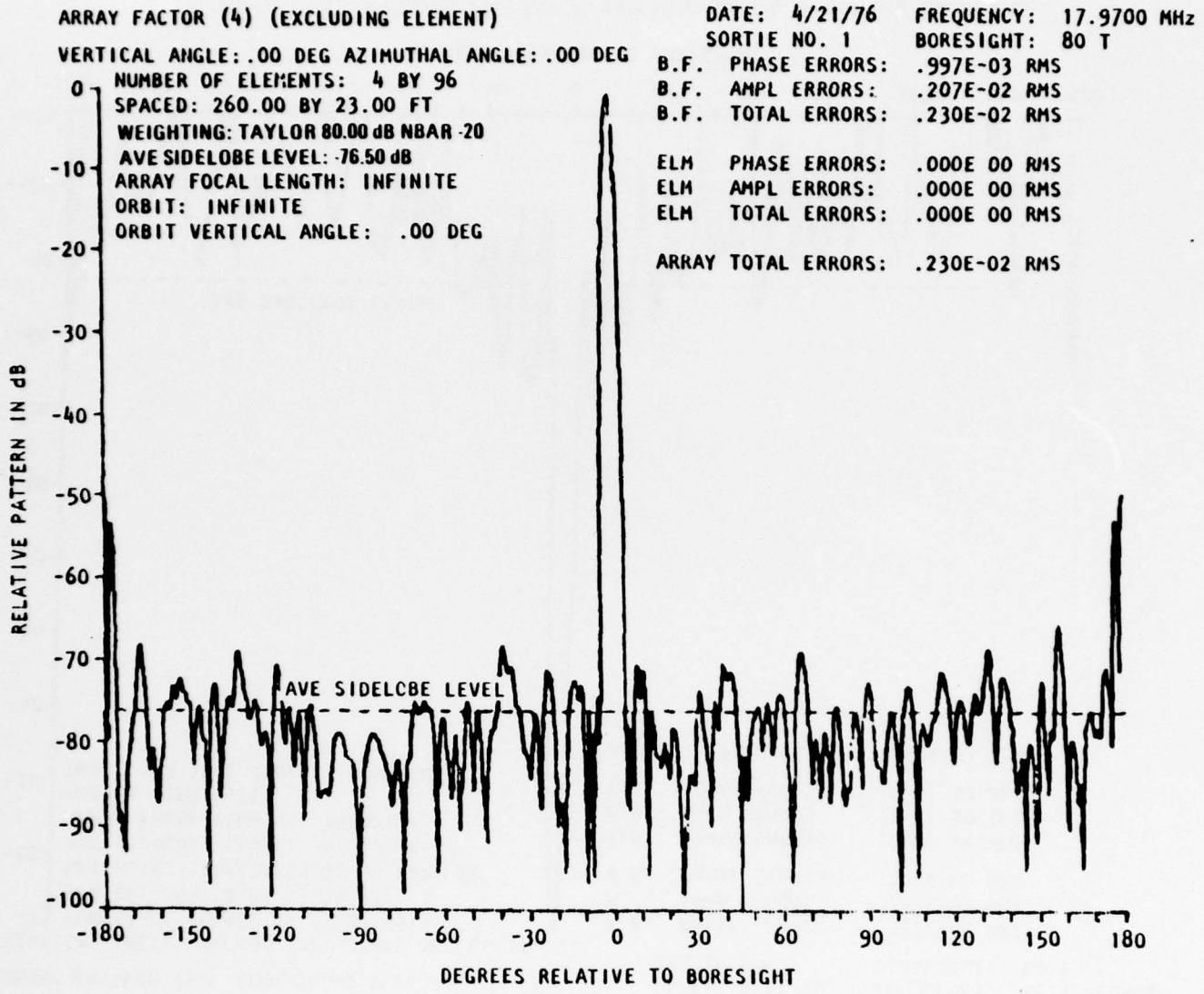


Figure 55. Pattern Plot for Array Factor (4) (Excluding Element)

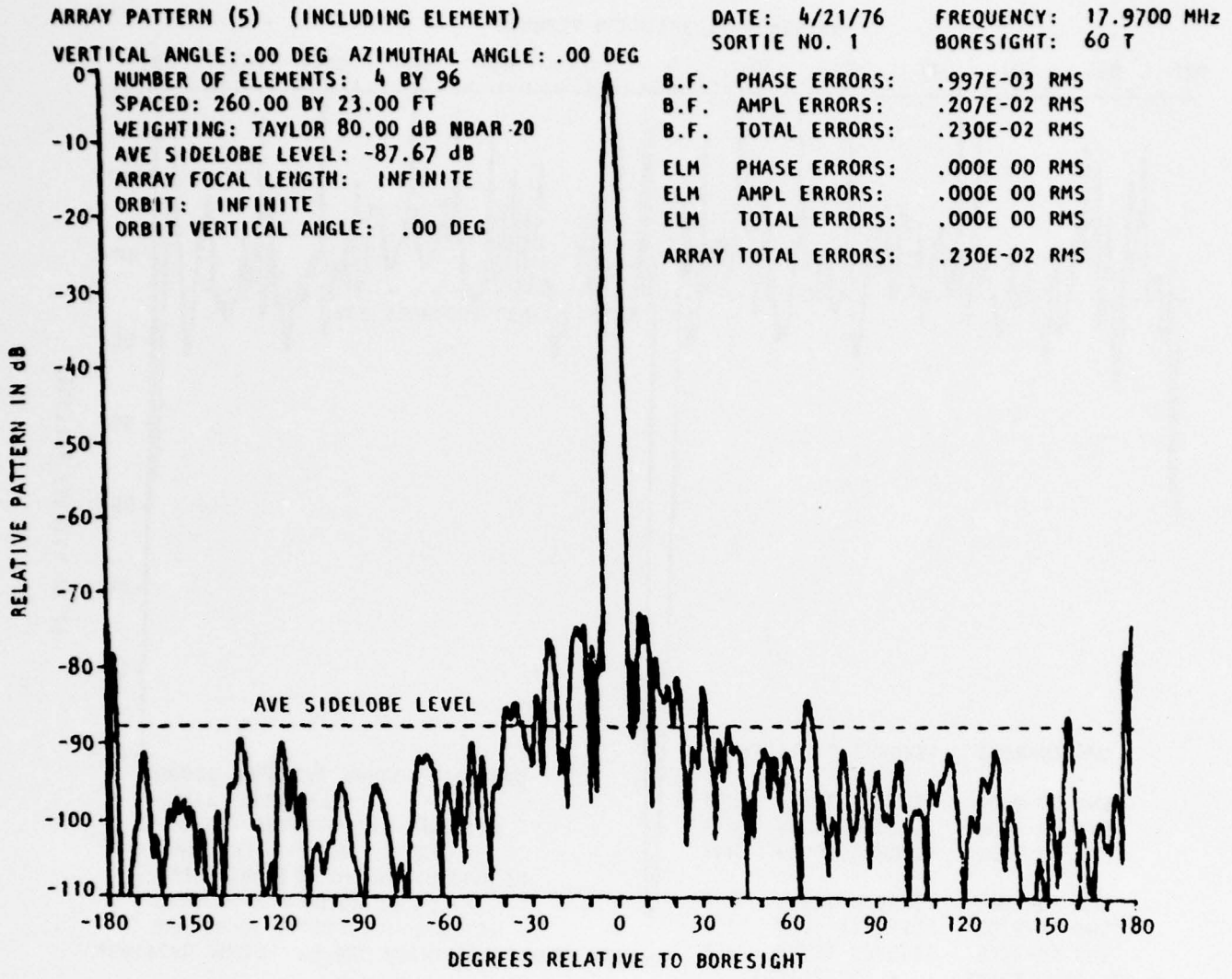


Figure 56. Pattern Plot for Array Pattern (5) (Including Element, Average Sidelobe Level -87.67 dB)

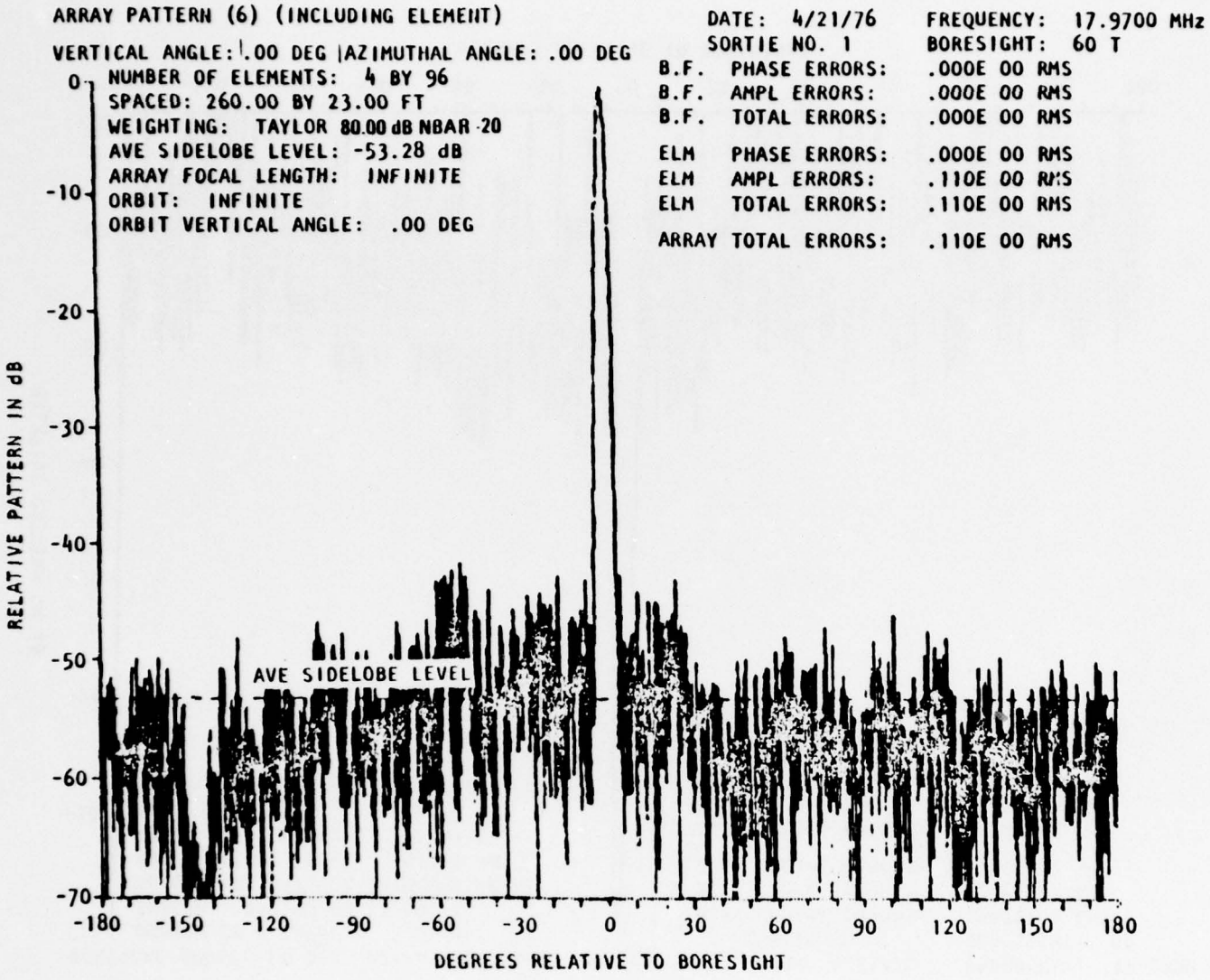


Figure 57. Pattern Plot for Array Pattern (5) (Including Element, Average Sidelobe Level -53.28 dB)

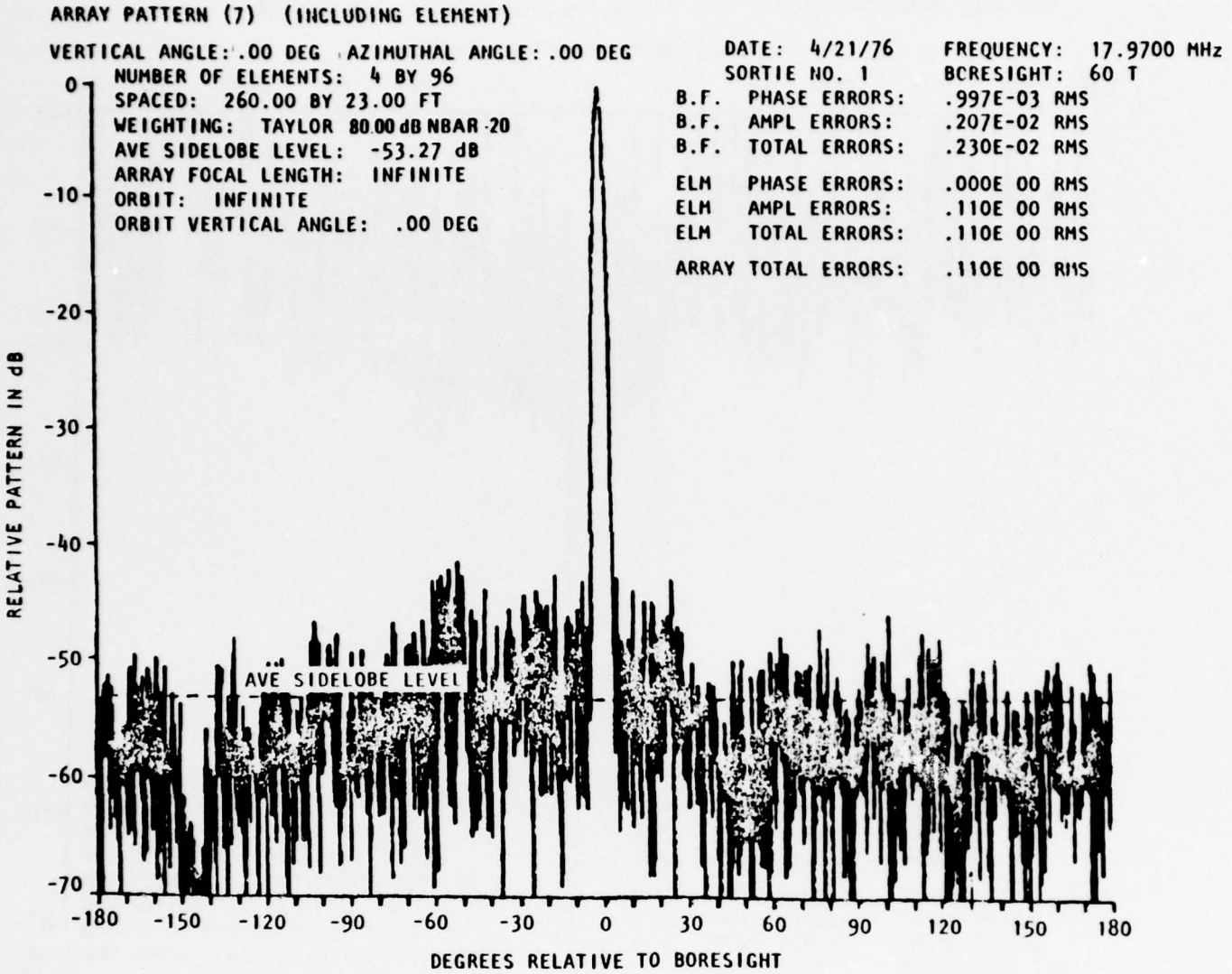


Figure 58. Pattern Plot for Array Pattern (7) (Including Element)

2.7 HISTOGRAM

Due to the problems encountered with the Raytheon APE equipment that was used to record the flight test data, the program HISTGRM was written. The problem encountered was that bit 13 of the analog-to-digital converter (ADC) would not always function. When it failed, the bit was also zero. Since the ADC is a two's complement converter, this causes severe distortion of the input signal, making the data unuseable.

The program, in order to check the quality of the data prior to use,

a. reads the data tape,

b. unpacks the 32-bit computer word into two 16-bit complex data words,

c. checks each bit of the 16-bit words to determine how many times the bit is a '1', and

d. provides the percentage of time each bit is ON for the tape.

Since the data should not contain a dc value, the distribution should be 50 percent. A bit not functioning disrupts this distribution, as does noise in the lower order bits.

To run this program, the user signs on the time-share terminal (as in Paragraph 2.5.9), except the assign is to file HISTGRMR. This causes the immediate execution of the load module HISTGRM, which was made using program HISTGRM.

The program HISTGRM, stored in the file of the same name, performs the following six functions.

a. Provides proper job card parameters
b. Makes the assignments for the tapes
c. Prints listings on the line printer
d. Compiles the Fortran source program
e. Loads the program binary and external files
f. Creates a permanent load module called HISTGRM:

2.8 BIT CORRECTION

An effort was started in an attempt to salvage the corrupted data tapes identified by the histogram programs. This effort was based upon the following assumptions about the data and signal.

a. Is a sine wave of 30 to 70 Hz

b. Has little or no dc value

c. Has no rapid amplitude fluctuations

d. Has at least a 50-dB signal-to-noise ratio (SNR)

The general approach was to consist of the following steps.

Step 1. Repeat the histogram check on each block of data read from the data tapes to determine if the problem is present during this time. If a problem is not present, no action is taken, and a new block of data input.

Step 2. If the bit 13 distribution deviates by a threshold value, from 50 percent, the data words are masked so that all words have the problem, not just a few.

Step 3. Match the data zero crossings with a calculated sinewave by estimating the frequency and maximum amplitude.

Step 4. Make a decision whether bit 13 is to be on or off, using the estimated frequency and amplitude, and set the bit accordingly. No effort is intended to correct any other bit.

The initial tests indicated the following limitations on the data assumptions.

a. The frequency assumption is adequate.

b. The dc value assumption is not useable; therefore, an effort to remove a dc bias would have to be made.

c. The power fluctuations in the towed transmitter used in the flight tests make this assumption suspect, but sufficiently accurate to proceed with initial corrections, at least.

d. The SNR assumption is quite accurate, as the SNR is set by the signal from the towed transmitter, and the SNR remains high even when the signal is in pattern nulls.

This effort was not finished due to contract termination. All notes and preliminary programs were delivered in place, as the customer indicated a desire to make an attempt to complete the effort.

2.9 DISCRETE U-SPACE TRANSFORM

The normal transform from element coefficients of a linear array to a spatial pattern is given by

$$E_{(\theta, \phi)} = \frac{1}{N} \sum_{n=\phi}^{N-1} C(n)\, e^{-i\,\delta(n, \theta, \phi)} \tag{25}$$

where:

$$\delta(n, \theta, \phi) = \frac{2\pi S n}{\lambda} (\sin(\theta) \cos(\phi) - \gamma) \tag{26}$$

$C(n)$ = amplitude coefficient of the n^{th} port

S = element spacing

λ = wavelength

θ = azimuthal angle-of-arrival

ϕ = vertical angle-of-arrival

γ = $\sin(\theta s)\ \cos(\Phi s)$

θs and Φs = the array steering angles

This transform yields the spatial pattern contained only in the "visible" region of space. In addition, it produces non-uniform sampling of the (θ, ϕ) space, thereby making inverse transformations difficult or impossible, as the total pattern is not represented.

To make inverse transformations of arbitrary patterns to aperture coefficients, the following transform was derived and programmed.

To obtain uniform sampling for forward and inverse transformations, the transformation will be made into U-space, as opposed to the (θ, ϕ) space.

The space factor for sampled arrays is given by

$$E_{(\theta, \phi)} = \left| \frac{\sin \frac{N\pi S}{\lambda}\left[(\sin\theta \cos\Phi - \gamma)\right]}{\sin \frac{\pi S}{\lambda}\left[(\sin\theta \cos\Phi - \gamma)\right]} \right| \tag{27}$$

Therefore, making

$$u = \frac{S}{\lambda} (\sin\theta \cos\Phi - \gamma) \tag{28}$$

Substituting into Equation (25) the space factor becomes,

$$E_{(u)} = \left/ \frac{\sin (N\pi u)}{\sin (\pi u)} \right/ \tag{29}$$

The nulls of $E_{(u)}$ are found when $E_{(u)} = 0$, which occurs when

$$N\pi u = k\pi \tag{30}$$

where k is an integer $(0, 1, 2, 3, \ldots K_{max})$.

Thus, the term $\frac{N\pi S}{\lambda} (\sin\theta \cos\phi - \gamma)$ from Equation (27) is equal to $k\pi$, and rearranging gives

$$(\sin\theta \cos\phi - \gamma) = \frac{k\lambda}{NS} \tag{31}$$

Substituting Equation (31) into Equation (26), it is found that

$$\delta(n, \theta, \phi) = \frac{2\pi kn}{N} \tag{32}$$

Thus, Equation (25) becomes the desired transform,

$$E_{(k)} = \frac{1}{N} \sum_{n=0}^{N-1} C_{(n)} e^{-i \frac{2\pi kn}{N}} \tag{33}$$

and the inverse transform is

$$C_{(n)} = \sum_{k=K_{min}}^{k=K_{max}} E_{(k)} \, e^{i \frac{2\pi k n}{N}} \tag{34}$$

Using

$$k = \frac{I}{M}, \quad \text{where } 0 \leq I < M \tag{35}$$

and moving the array phase center to the physical array center, Equations (34) and (35) become

$$E_{(I)} = \frac{1}{N} \sum_{n=1}^{N} C_{(n)} \, e^{-i \frac{\pi I}{M} \left(1 - \frac{2n}{N}\right)} \tag{36}$$

and

$$C_{(n)} = \sum_{I=MK_{min}}^{I=MK_{max}} E_{(I)} \, e^{-i \frac{\pi I}{M} \left(1 - \frac{2n}{N}\right)} \tag{37}$$

(in steps of M)

The range of k, using the Nyquist criteria,

$$K_{min} \leq k \leq K_{max} \tag{38}$$

where:

$K_{min} = -N/2$

$K_{max} = N/2$

The range equality (38) defines the limits that include the "invisible" space pattern. The range of k

$$K_{minvs} \leq k \leq K_{maxvs} \tag{39}$$

defines the range of visible space.

The values of K_{minvs} and K_{maxvs} are found using Equation (31) and the following equality,

$$-1 \leq \sin\theta \cos\phi \leq 1 \tag{40}$$

as

$$K_{minvs} = -\frac{NS}{\lambda}(1 + \gamma) \tag{41}$$

and

$$K_{maxvs} = \frac{NS}{\lambda}(1 - \gamma) \tag{42}$$

The transform to the normal spatial pattern in the (θ, ϕ) phase is

$$E_{(\theta,\phi)} = E_{(I)} \tag{43}$$

where

$$\theta = \arcsin\left\{\frac{1}{\cos\phi}\left(\frac{I\lambda}{MNS} + \gamma\right)\right\} \tag{44}$$

where:

$$MK_{minvs} \leq I \leq MK_{maxvs}$$

These transforms were used to attempt to develop super gain patterns (e.g., narrow beams); in addition, an attempt was made to retain low sidelobes.

This effort, while producing some super gain patterns (with 15 to 20 dB sidelobes) and the corresponding aperture coefficients, was terminated due to the demands of the Dexter array design and implementation.

2.10 MEASUREMENT STORAGE

Two major measurement programs were undertaken at the Dexter, New York, site.

a. Phase measurements on the signal cables from the field to the relay logic units

b. Element elevation data referenced to a local datum plane

To record this data, two programs were written for the PDP-11/40 system. The programs were designed to input the data points and store in disk storage. In addition, the programs could be used to get complete listings of the raw data or refined data.

These measurement data are stored on a disk at Dexter, New York, labeled "Measured Data and Programs for Entering and Updating."

Section 3

SUMMARY AND CONCLUSIONS

From 1 May 1974 to 30 June 1976, GTE Sylvania Incorporated, ESG-WD, provided engineering field services in support of the RADC experimental HF FM/CW backscatter system and the design and installation of a computer-controlled low sidelobe design array. These experimental/theoretical investigations were primarily concerned with the design of the computer control logic and software design and implementation on the PDP-11/40 computer system, and development of a computer model of the Dexter array that would incorporate the measured antenna patterns and error functions.

The computer-controlled functions, as far as implemented, operated as expected. Due to program and contract termination prior to completing the units, no extensive testing was done on any part of the system (except the element patterns, which were measured and reported internally by RADC). The limited preliminary testing of breadboards and concepts indicated that the computer-controlled trimming units and the CAL1 measuring system could have set the beamformer system to the desired values.

There is no data concerning the time-dependent variations to ascertain any estimate of the recalibration period required for resetting the system to precalibrated parameters. However, all critical components were designed to be as identical as possible and their environment to be the same. Thus, it is reasonable to assume that, if component drift (particularly the signal cables in the field) were experienced, the drift would be common, thus a relative difference maintained near zero.

It is reasonable to conclude the beamformer and computer controls would have met the specifications required to meet the design goals. It will depend upon the results of the element flight test data and reduction to conclude if the array, including the element effects, would have met the design goals.

METRIC SYSTEM

BASE UNITS:

Quantity	Unit	SI Symbol	Formula
length	metre	m	...
mass	kilogram	kg	...
time	second	s	...
electric current	ampere	A	...
thermodynamic temperature	kelvin	K	...
amount of substance	mole	mol	...
luminous intensity	candela	cd	...

SUPPLEMENTARY UNITS:

plane angle	radian	rad	...
solid angle	steradian	sr	...

DERIVED UNITS:

Acceleration	metre per second squared	...	m/s
activity (of a radioactive source)	disintegration per second	...	(disintegration)/s
angular acceleration	radian per second squared	...	rad/s
angular velocity	radian per second	...	rad/s
area	square metre	...	m
density	kilogram per cubic metre	...	kg/m
electric capacitance	farad	F	A·s/V
electrical conductance	siemens	S	A/V
electric field strength	volt per metre	...	V/m
electric inductance	henry	H	V·s/A
electric potential difference	volt	V	W/A
electric resistance	ohm		V/A
electromotive force	volt	V	W/A
energy	joule	J	N·m
entropy	joule per kelvin	...	J/K
force	newton	N	kg·m/s
frequency	hertz	Hz	(cycle)/s
illuminance	lux	lx	lm/m
luminance	candela per square metre	...	cd/m
luminous flux	lumen	lm	cd·sr
magnetic field strength	ampere per metre	...	A/m
magnetic flux	weber	Wb	V·s
magnetic flux density	tesla	T	Wb/m
magnetomotive force	ampere	A	...
power	watt	W	J/s
pressure	pascal	Pa	N/m
quantity of electricity	coulomb	C	A·s
quantity of heat	joule	J	N·m
radiant intensity	watt per steradian	...	W/sr
specific heat	joule per kilogram-kelvin	...	J/kg·K
stress	pascal	Pa	N/m
thermal conductivity	watt per metre-kelvin	...	W/m·K
velocity	metre per second	...	m/s
viscosity, dynamic	pascal-second	...	Pa·s
viscosity, kinematic	square metre per second	...	m/s
voltage	volt	V	W/A
volume	cubic metre	...	m
wavenumber	reciprocal metre	...	(wave)/m
work	joule	J	N·m

SI PREFIXES:

Multiplication Factors	Prefix	SI Symbol
1 000 000 000 000 = 10^{12}	tera	T
1 000 000 000 = 10^{9}	giga	G
1 000 000 = 10^{6}	mega	M
1 000 = 10^{3}	kilo	k
100 = 10^{2}	hecto*	h
10 = 10^{1}	deka*	da
0.1 = 10^{-1}	deci*	d
0.01 = 10^{-2}	centi*	c
0.001 = 10^{-3}	milli	m
0.000 001 = 10^{-6}	micro	μ
0.000 000 001 = 10^{-9}	nano	n
0.000 000 000 001 = 10^{-12}	pico	p
0.000 000 000 000 001 = 10^{-15}	femto	f
0.000 000 000 000 000 001 = 10^{-18}	atto	a

* To be avoided where possible.

MISSION
of
Rome Air Development Center

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C^3) activities, and in the C^3 areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

*U.S. GOVERNMENT PRINTING OFFICE: 1977-714-025/139